Heinrich-Heine-Universität Düsseldorf

Analysis of Multiplexing for Multipartite Quantum Repeaters

Inaugural dissertation

presented to the Faculty of Mathematics and Natural Sciences of the
Heinrich-Heine-Universität Düsseldorf
for the degree of

Doctor of Natural Sciences (Dr. rer. nat.)

by

Julia Alina Kunzelmann

from Haan

Düsseldorf, June 2025

From the Institute for Theoretical Physics III
at the Heinrich-Heine-Universität Düsseldorf

Published by permission of the
Faculty of Mathematics and Natural Sciences at
Heinrich-Heine-Universität Düsseldorf

Supervisor: Prof. Dr. Dagmar Bruß
Co-supervisor: PD Dr. Hermann Kampermann

Date of the oral examination: 14.07.2025

Declaration

Ich versichere an Eides statt, dass die Dissertation von mir selbstständig und ohne unzulässige fremde Hilfe unter Beachtung der “Grundsätze zur Sicherung guter wissenschaftlicher Praxis an der Heinrich-Heine-Universität Düsseldorf” erstellt worden ist.

Düsseldorf, 02. Juni 2025

Julia Alina Kunzelmann

To my parents, Heike and Frank,
for your unwavering support, love, and belief in me throughout this journey.

Abstract

Quantum repeaters are central components in quantum networks placed between two parties to divide the total communication distance into shorter sections. Through local operations, the quantum repeaters enable the end nodes to be entangled without them interacting directly with each other. Holding an entangled state, the parties can perform quantum key distribution to share a secret key. To increase the entanglement generation rate, and therefore, the secret key rate between the parties, memories can be added to the quantum repeater. Those memories allow for parallelizing the entanglement distribution attempts, a technique called multiplexing.

In many real-world applications, multipartite entanglement between N parties is needed. Therefore, in this work, we generalize the concept of bipartite multiplexing to multipartite multiplexing in multipartite quantum repeaters, in the following called quantum routers. In a first step, we investigate the underlying multipartite matching problem, which resembles a problem already known in graph theory. The matching defines how the qubits are optimally combined so that the resulting entangled states are of high fidelity. The fidelity is influenced by the storage time – the longer a qubit is stored, the more the fidelity drops. By considering different matching strategies, we were able to maximize the fidelity and the secret key rate. Our results show that it is optimal to combine the freshest qubits with the highest fidelities first. Moreover, qubits whose fidelity has already dropped under a certain threshold should be removed from the memories. We further analyze how the entanglement generation rate depends on the size of the quantum router network. Therefore, we derive a general expression that describes the entanglement generation rate as a function of the number of parties and the available memories per party.

Our results provide a basis for planning and scaling quantum router networks with a central quantum router. They make it possible to estimate in advance which entanglement rates can be achieved for a given number of parties and how these can be increased by adding quantum memories. At the same time, the matching strategies we investigate offer the possibility of minimizing the influence of quantum memories and maximizing the secret key rate.

Zusammenfassung

Quantenrepeater sind zentrale Komponenten in Quantennetzwerken, die dazu dienen, verschränkte Zustände über größere Distanzen zwischen zwei Endknoten, den Parteien, zu verteilen. Die Quantenrepeater sind so zwischen den Parteien positioniert, dass sie die Gesamtstrecke in kürzere Abschnitte unterteilen. Durch lokale Operationen ermöglichen sie eine Verschränkung der beiden Endknoten, ohne dass diese direkt miteinander interagieren. Mithilfe dieser Verschränkungen können die Parteien einen sicheren Quantenschlüsselaustausch durchführen. Um mehr verschränkte Zustände pro Versuch zu erzeugen, können dem Repeater Quantenspeicher hinzugefügt werden. Diese ermöglichen die Parallelisierung der Versuche zur Verschränkungsverteilung, ein Verfahren, das als Multiplexing bezeichnet wird.

In vielen praktischen Anwendungen wird eine multipartite Verschränkung zwischen N Parteien benötigt. In dieser Arbeit verallgemeinern wir daher das Konzept des bipartiten Multiplexings auf das multipartite Multiplexing in multipartiten Quantenrepeatern, im folgenden Quantenrouter genannt. Als ersten Schritt untersuchen wir das dabei entstehende Problem des multipartiten Matchings, das in ähnlicher Form bereits aus der Graphentheorie bekannt ist. Das Matching bestimmt, wie die Qubits im Quantenrouter optimal kombiniert werden, sodass die resultierenden verschränkten Zustände möglichst hohe Zustandsgüten aufweisen. Diese Güte wird durch die Speicherzeiten der Qubits beeinflusst – je länger ein Qubit gespeichert ist, desto stärker nimmt seine Güte ab. Durch verschiedene Matchingstrategien konnten wir die Güte und die daraus resultierende geheime Schlüsselrate maximieren. Unsere Ergebnisse zeigen, dass es optimal ist, in jeder Runde bevorzugt die neuesten Qubits mit höchster Güte zu kombinieren. Zudem sollte ein Qubit, dessen Güte unter einen bestimmten Schwellwert gefallen ist, aus dem Speicher gelöscht werden. Zusätzlich untersuchen wir, wie sich die Rate der Verteilung verschränkter Zustände mit der Größe des Netzwerkes verändert. Hierfür leiten wir eine allgemeine Formel her, welche die Rate in Abhängigkeit von der Anzahl an Parteien und der verfügbaren Speicher pro Partei beschreibt.

Unsere Ergebnisse bieten eine Grundlage für die Planung und Skalierung von Quantenrouternetzwerken mit einem zentralen Quantenrouter. Sie ermöglichen es,

im Voraus abzuschätzen, welche Raten für die Verteilung verschränkter Zustände bei gegebener Anzahl an Parteien erreichbar sind und wie sich diese durch das Hinzufügen von Quantenspeichern steigern lässt. Gleichzeitig bieten die von uns untersuchten Matchingstrategien die Möglichkeit, den Einfluss der Quantenspeicher zu minimieren und die geheime Schlüsselrate zu maximieren.

Acknowledgement

Exploring the depths of quantum networks and quantum information theory has been a deep intellectual journey. I am sincerely grateful for the opportunity to take this step and for the support of those who accompanied me along the way. Before delving into this fascinating topic, I would like to take a moment to thank all the people who made this journey possible.

First of all, I would like to thank my supervisor, Dagmar Bruß, for the possibility to be part of her group and for all the scientific and non-scientific support during these years. Your door was always open for helpful discussions and advice.

I am equally grateful to my co-supervisor, Hermann Kampermann, who helped me through this time with supportive and valuable discussions.

A special thanks to my co-authors for the great teamwork and the many inspiring conversations along the way.

I would also like to take this opportunity to thank all my colleagues in the group, many of whom became friends over time. I am thankful for all the inspiring discussions, the conferences we attended, and the many enjoyable moments we shared outside of work. This journey was not always easy, but with the group's support, it was always possible to move forward and find a way. Thank you, Sarnava, Lucas, Thomas, Federico, Giacomo, Chandan, Gláucia, Nikolai, Justus, Ghi, Raphael, Varun, Anton, Pedro, Yien, and Carolin. I especially would like to thank Raphael, Ghi, Justus, Yien, Christian, Varun, and Anton for proofreading this thesis.

Many thanks also to all other members of this institute, particularly Jens Bremer, Claudia Glowacki, and Cordula Hoffjan, for helping me with all the bureaucratic and technical tasks.

I want to extend my heartfelt thanks to all my other friends. Their steady support, patience, and ability to lift my spirits after a long, exhausting working day made all the difference, and I am truly grateful for your presence in my life.

The greatest thanks go to my family—to my parents, Heike and Frank, and my brother, Tobi. Without you, this journey would have been impossible. Thank you for your unconditional support over all these years. You have shaped me into the person I am today, constantly encouraging me to grow beyond my limits. My life would not be the same without your love and unwavering support.

Last but not least, my endless gratitude goes to Christian for always being by my side and supporting me, no matter what. I am grateful for every moment with you – and for all those yet to come.

CONTENTS

CHAPTER 1

INTRODUCTION

With the development of quantum mechanics in the early 20th century, the way to modern physics was paved. This year, we are celebrating 100 years of quantum mechanics – 100 years in which new theories and new physics research fields were investigated. From the development of the model of the atom by Niels Bohr in 1913 [Boh13] to the discovery of quantized spin in atoms in 1922 [GS22], and the quantum description of solids in 1929 [Blo29], many discoveries and theories have reshaped modern physics.

The first introduction of the concept of entanglement by A. Einstein, B. Podolsky, and N. Rosen in 1935 [EPR35] led to many controversial discussions. With the introduction of the Bell inequalities in 1964 [Bel64], the discussions shifted more toward experimental considerations. John S. Bell argued that the theory of hidden variables implies mathematical constraints on the measurements performed on two separate entangled particles. These constraints are known as the Bell inequalities. Entanglement was first demonstrated experimentally in a Bell test in 1972 [FC72]. In that experiment, S. Freedman and J. Clauser were the first to demonstrate the violation of Bell's inequality using photons from excited calcium atoms. In 2022, J. Clauser, together with A. Aspect and A. Zeilinger received the Nobel Prize in Physics for the proof of Bell's inequality, which demonstrates quantum entanglement, and other groundbreaking discoveries in *quantum information theory.*

Quantum information theory is a field of research that developed from quantum mechanics. Due to the properties of quantum particles, such as superposition (demonstrated in Schrödinger's cat thought experiment by E. Schrödinger in 1935[Sch35]) and quantum entanglement, these are suitable for information theory. Analogous to classical bits, quantum bits (qubits) can be used to store or transmit information. The no-cloning theorem for arbitrarily unknown quantum states introduced fundamental implications for quantum computing and cryptography in 1982 [WZ82]. Two years later, the first protocol for quantum cryptography was presented [BB84]. Using this protocol, two parties can exchange a secret key that is inherently secure from eavesdroppers. Just one year later, in 1985, D. Deutsch showed that it is possible to simulate any physical system with a universal quantum computer and highlighted tasks, for which such a device outperforms classical computers [DEJ$^+$96]. Further quantum algorithms were introduced in the 1990s. In 1994, Shor developed an algorithm for prime factorization that can be used to break classical cryptosystems (public key cryptography) [Sho94].

To date, there have been many further discoveries in the field of quantum information theory. In this work, we focus on quantum networks. A network is an association of end nodes, the so-called users, who exchange information over long distances. In the following, we address the question of how the users (or parties) can share entangled states over long distances within a network.

1.1 Motivation and related work

Quantum entanglement is one of the main resources needed to share or transmit information or secret keys (quantum key distribution) between the parties. Distributing these entangled states over longer distances is challenging. Due to interactions with the environment, a quantum system evolves with time and, therefore, loses information. A distance of about 150-200 km can be bridged by, for example, sending photons through an optical fiber without using advanced technologies [HRP$^+$06, SWV$^+$09].

For a fixed distance d between the parties, a strict upper bound on the achievable secret key rate in quantum key distribution (QKD) is given by the *PLOB bound* named after its inventors Pirandola, Laurenza, Ottaviani, and Banchi [PLOB17]:

$$K_{PLOB} = -\log(1-\eta), \tag{1.1}$$

with transmittance $\eta = 10^{-\alpha d/10}$ being the fraction of photons that are sent through the fiber successfully. For a commercial optical fiber used with light of a wavelength of 1550 nm, for the attenuation coefficient α it holds $\alpha = 0.2$ dB/km. The PLOB bound sets the limit on the rate of quantum key distribution that is achievable over a repeaterless lossy quantum channel, such as an optical fiber. This limit is independent of the QKD protocol, i.e., all protocols that only use direct transmission (so no memories and no entanglement swapping) cannot beat this bound. To get secret key rates beyond this limit, new strategies have to be developed.

One strategy to increase the distance between the parties while beating the PLOB bound is given by advanced technologies such as twin-field quantum key distribution [LYDS18]. Here, the parties do not directly send qubits (in the form of photons) to each other, but instead, send weak coherent pulses to a central station that performs interference measurements. Using twin-field QKD, the distance between the parties can be remarkably increased above 500 km [ZLJY23, LZJ$^+$23]. Another strategy, which is experimentally more challenging, is the use of quantum repeaters [BDCZ98]. These intermediate stations are put between the parties to divide the distance into smaller segments, thus overcoming the PLOB bound. Unlike classical amplifiers, a quantum repeater does not amplify the signal within the channel. According to the no-cloning theorem, copying an arbitrary unknown quantum state is impossible. Therefore, amplifying an incoming signal (quantum bit) in a classical manner is not feasible. Instead, long-distance entangled links are generated by performing local measurements (entanglement swapping) at the stations. As a result, end nodes share entangled links even without previous interactions. Compared to the intermediate station in twin-field QKD, a quantum repeater is a more complex device that needs quantum memories to store the incoming quantum bits. In addition to the quantum memories, components to perform the entanglement swapping are needed. The main challenges for a quantum repeater to tackle are long coherence times, fault-tolerance, and low-loss components. To deal with these requirements, three generations of quantum repeaters exist:

- **1st generation.** Quantum repeaters of the first generation work with entanglement swapping at the central station and additional purification [BDCZ98]. Due to the purification, error tolerance is given.

- **2nd generation.** For these quantum repeaters, purification is replaced by quantum error correction [JTN$^+$09]. Additional entanglement swapping still needs to be performed. Quantum repeaters of the second generation are faster and more scalable. However, these repeaters still need quantum memories.

- **3rd generation.** Here, no quantum memories are needed, and only quantum error correction is performed at the quantum repeater. This method does not require two-way communication, so it is faster and suitable for networks. In contrast, fault-tolerant quantum gates and high-rate quantum error correction are required here.

An overview of all three generations of quantum repeaters is given in [MLK$^+$16].

Each generation has advantages and disadvantages. In general, the quantum repeater of the third generation is technologically the most advanced, since it is faster and more scalable. Nevertheless, for realization, fully fault-tolerant and high-fidelity components are required, which are not yet widely available. So far, the analysis remains theoretical, and no experimental demonstration has been made. Quantum repeaters of the second generation use quantum error-correction, which leads to faster transmissions. To reach this, higher gate fidelities are also required, as well as quantum memories with long coherence times. Also, no full second-generation quantum repeater has been demonstrated so far. In this work, we focus on first-generation quantum repeaters. Those are ideal devices for early-stage networks and are good for proof-of-principle demonstrations. Since an experimental roadmap was given by Duan et al. in 2001 [DLCZ01] (DLCZ proposal, named after its authors), several experimental demonstrations have been realized. In 2005, the first demonstration of a quantum memory for a single photon was made [CMJ$^+$05], and entanglement between light and matter could be generated [CDRF$^+$05]. After further significant steps were achieved, memory-based quantum communication was demonstrated, achieving a four-fold increase in the secret key rate compared to the loss-equivalent direct-transmission method [BRM$^+$20].

One of the challenges that still has to be tackled is the low entanglement generation rate. In another demonstration of a quantum repeater node using trapped ions [KCM+23], for example, the achieved rate (entangled pairs per second) is about 1-10 Hz. Assuming that, due to post-processing, only 10% of the entangled pairs generate a secret bit for the key, it takes between ~ 4 and ~ 40 minutes to generate a key of length 256 bits. With this key length, only short messages can be encoded. In real-world applications, longer keys are needed, thus significantly increasing the key generation time. Different strategies to increase the entanglement generation rate have been proposed [BPvL11, SSdRG11]. One method, first introduced in [CJKK07], is the multiplexing. In spatial multiplexing, several memories are put in parallel, so that multiple independent attempts of distributing entangled states can be made per communication round. This leads to an increase in the entanglement generation rate and a reduction of the waiting times of the qubits in the memories, which are major bottlenecks in real-world quantum networks. Collins et al. [CJKK07] show that by introducing m memories (each distributing entangled states with probability p), the total success probability p_{tot} for entanglement distribution in a single attempt becomes

$$p_{tot} = 1 - (1 - p)^m \tag{1.2}$$

which can be approximated by mp for small p, i.e., $p \ll 1$. Compared to a simple quantum repeater without spatial multiplexing, for which the success probability is simply p, a linear speed-up in m is reached. Consequently, the waiting time drops by a factor m. Further proposals of quantum repeaters using memory multiplexing are given, for example, in [SdRA+07, SSdRG11]. Demonstrations of a quantum repeater following the DLCZ proposal and integrating spatial multiplexing have been realized in recent years. In [PJC+17], the authors use acousto-optic deflectors to individually address 225 spatial cells. They demonstrate the independent access and control over many memory elements. In [LZW+21], multiple atomic memory cells are used to demonstrate spatial multiplexing. The authors improve the entanglement generation rate and reduce waiting times by parallelizing quantum memory channels using four independently operated atomic quantum memory cells.

A question that arises when dealing with spatial memory multiplexing is how to combine the quantum memories in a manner that results in an optimal entanglement generation rate. Due to the interaction of qubits with the quantum memories during storage, the fidelity of the qubits degrades, which in turn reduces the secret key rate. Depending on the platforms that are used to realize the setup, different memory lifetimes between 1 ms [LZW$^+$21] and several minutes [WUZ$^+$17] are achievable. Motivated by the fact that long-range connections are experimentally challenging [GMM$^+$24], Abruzzo et al. introduce the so-called finite-range multiplexing protocol that takes into account that all-to-all connections might be demanding to realize experimentally [AKB14a].

In real-world applications, it is often not sufficient to distribute entangled states between two parties only. Instead, multipartite entanglement between N parties is needed – for example, in secret sharing [HBB99], leader election [ABDR04], or conference key agreement [CL08]. Therefore, we consider multipartite quantum repeaters and generalize the concept of (finite-range) multiplexing to the multipartite setup in this work. We focus on multipartite quantum networks with one central station that is placed between the N end nodes. As the central station is used to "route" (distribute) entangled states among all parties, we call this station the quantum router. Similar network structures are analyzed in other works about multipartite quantum routers (also called quantum switches in the literature) [NVGT20, NVGT22, VGNT21a, MEW23]. In [BVK98], a generalization to the multipartite entanglement swapping is given, thus laying the groundwork for multipartite entanglement distribution in quantum networks. A two-dimensional quantum repeater protocol using an all-photonic framework to generate multipartite entanglement over long distances is presented in [LFL$^+$23a]. The performance of the entanglement generation (i.e., entanglement generation rate and fidelity) via a central node in a star-shaped network is analyzed in [ARW23]. The distribution of the GHZ state is either realized via a factory node or a 2-switch, in which the GHZ state is created stepwise from all the Bell states. The distribution of an entangled state between a subset of all parties has been analyzed numerically [CKD$^+$21] and also analytically via Markov chains [VGNT21b]. It is also possible to perform conference key agreement via a central quantum router without integrating quantum memories [LFL$^+$23b].

Compared to previous work about quantum routers, we focus on integrating quantum memories to perform memory multiplexing within the quantum router. Due to the parallelization of the entanglement distribution, we increase the entanglement generation rate. Additionally, we analyze various strategies for choosing the qubits for the multipartite entanglement swapping depending on the fidelities of the qubits. This allows us to minimize the influence of the quantum memories on the fidelity of the qubits in order to keep the secret key rate as high as possible. Note that our setup can be seen as a generalization of the measurement-device-independent quantum key distribution protocol with quantum memories from [AKB14b] to more than two parties.

1.2 Overview of results

In this work, we generalize the finite-range multiplexing from the bipartite quantum repeater to a multipartite quantum router modeled as a star graph connecting N parties. We formalize the underlying quantum router matching problem that must be solved to enable optimal performance in choosing the qubits for the entanglement swapping in the quantum router. Specifically, we show that this matching problem as a decision problem, analogous to the well-known N-dimensional matching, is $\mathcal{NP}$-complete. We further identify special cases that admit efficient polynomial-time solutions and provide concrete algorithms for solving them. For the general unweighted maximum quantum router matching, we propose an approximation algorithm that solves the problem, but does not always guarantee finding a matching with maximum cardinality.

Building on this theoretical foundation, we analyze the influence of the quantum memories on the secret key rate and investigate various multiplexing strategies for optimizing the secret key rate. For this purpose, we use the algorithms for the quantum router matching that we derive in the prior work. We show that the best strategy is always to use the freshest qubits first (with the highest fidelities) and remove qubits whose fidelities drop below a certain threshold – the fidelity cutoff. For this work, we develop a simulation of the multipartite quantum router in a star graph that models the distribution of multipartite entangled states among all parties and calculates the achievable rates (entanglement generation rate and secret key rate) based on the samples generated by the simulation. Compared to other simulators [CKD+21, WHW+24], we focus on the multipartite quantum router and especially the integration of multiplexing and the underlying matching problem.

Although quantum memories introduce decoherence over time – leading to an increase in the quantum bit error rates – they also raise the router rate, i.e., the rate at which entangled states can be distributed across all parties. In this thesis, we investigate the link between the router rate and both the number of parties N and the number of memories m per party. We derive an approximate formula capturing this relationship and demonstrate that the minimal rate achievable with two parties and a single memory each can be maintained even as the network scales to arbitrary sizes.

1.3 Thesis structure

The dissertation is organized as follows:

- In Chapter 2, we provide the fundamentals of quantum mechanics that are needed for further understanding of this work. We first introduce the postulates of quantum mechanics and the mathematical background for quantum information theory. We then introduce the concept of entanglement, quantum networks, and quantum key distribution.

- The proper definition and analysis of the underlying quantum router matching problem is given in Chapter 3. Some background from computer science about complexity and the matching problem in graphs is explained. We present our results about the complexity of the quantum router matching and give the algorithms for some special cases. Further, we introduce the approximation algorithm that solves the general unweighted maximum quantum router matching. This is based on our work [BGK+25] and [JAK25].

- In Chapter 4, we investigate the impact of the quantum memories when storing the qubits. We first give more information about the conference key agreement and derive formulas for the quantum bit error rates and the secret key rate needed for the considered star-shaped quantum router network. We describe the simulations that we perform to simulate the quantum router network and integrate different matching strategies. We discuss these strategies with respect to the achievable secret key rates and find optimal strategies that lead to maximal rates. These results are mainly based on [KKB24], while the algorithms for solving the underlying matching in the simulations are taken from [BGK+25].

- In Chapter 5, we give an approximation formula to calculate the achievable router rate in the stationary regime. Based on the theory of Markov chains that we further specify in this chapter, we derive the relation between the asymptotic router rate and the network size for different scenarios. Those results are based on [KTW+25].

CHAPTER 2

THEORETICAL BACKGROUND

About 100 years ago, contradictions regarding the microscopic nature of our universe arose in physics, leading to a new research field. At that time, physicists started to predict absurd theories [NC10] such as the 'ultraviolet catastrophe' leading to infinite energy [Ehr11], or electrons that could spiral into the nucleus. Theories that were based on the fundamentals of classical physics reached their limits. As a result, the modern theory of *quantum mechanics* was born. With high success, that theory was applied in many different research fields. One of the fields of modern quantum mechanics is the quantum information theory [Bru25]. Our work focuses on the aspects of quantum communication within quantum networks, which forms one possible application of quantum information theory.

To understand the basics of quantum mechanics, we start with the postulates, which were formulated in the early days of quantum mechanics. We further introduce the main topics of quantum information theory and quantum networks to guide the reader through the necessary background.

The topics presented here are covered in [NC10, HHHH09, JFZ22, GHZ89, VM14, AKB14a, LP09, MT09, BB84, MGKB20]. Additional inspiration has been drawn from [Wol21, Ten23, Abr14, Gra21].

2.1 Postulates of quantum mechanics

With the postulates of quantum mechanics, we give a relation between the physics and the mathematical description of quantum mechanics. The postulates were developed in the early days of quantum mechanics and still serve as the foundation of modern physics. This section provides an overview of the fundamental concepts required to understand the following work. The presentation of the postulates follows [NC10].

2.1.1 Quantum states

We now introduce the first postulate of quantum mechanics, which deals with the *Hilbert space* $\mathcal{H}$. It is the space in which quantum mechanics takes place. Here, we restrict ourselves to the finite-dimensional case.

Postulate 1.([NC10]) Associated to any isolated physical system is a complex vector space with inner product (that is, a Hilbert space) known as the *state space* of the system. The system is completely described by its *state vector*, which is a unit vector in the system's state space.

To understand the first postulate, some basic notations from linear algebra are needed. For now, we consider *isolated quantum systems*, i.e., we demand systems that do not interact with other systems or the environment. The state vector is usually given in *Dirac notation* where the vectors are denoted as a *ket* $|\psi\rangle$. Note that the symbol ψ is a label for the vector. It can be replaced by any other symbol. For the inner product, also the *dual space* $\mathcal{H}^*$ and the dual vectors $\langle\psi|$ are needed. Given a complex field $\mathbb{C}$, the dual space $\mathcal{H}^*$ is given by the set of all linear maps $\langle\psi| : \mathcal{H} \mapsto \mathbb{C}$ where the vectors $\langle\psi|$ are called *bra*. The dual vector $\langle\psi|$ relates to the state vector $|\psi\rangle$ via the *Hermitian conjugate* $(\cdot)^\dagger$:

$$(\alpha|\psi_1\rangle + \beta|\psi_2\rangle)^\dagger = \bar{\alpha}\langle\psi_1| + \bar{\beta}\langle\psi_2| \tag{2.1}$$

$$\left(\bar{\alpha}\langle\psi_1| + \bar{\beta}\langle\psi_2|\right)^\dagger = \alpha|\psi_1\rangle + \beta|\psi_2\rangle \tag{2.2}$$

for any complex numbers α, β and its complex conjugates $\bar{\alpha}, \bar{\beta}$. From that, the *inner product* is defined as follows:

Definition 2.1 *(Inner product). The mapping* $\langle\cdot|\cdot\rangle : \mathcal{H} \times \mathcal{H} \mapsto \mathbb{C}$ *takes two vectors* $|\psi\rangle, |\phi\rangle \in \mathcal{H}$ *as input and associates them with a complex number. It is called the inner product if it satisfies the following conditions:*

1. $\langle\psi|\phi\rangle = \overline{\langle\phi|\psi\rangle}$ *(conjugate symmetry) ,*
2. $\langle\psi|\psi\rangle \geq 0$ *with equality if and only if* $|\psi\rangle = 0$ *(positive definiteness),*
3. $\langle\psi| \sum_i \lambda_i \phi_i\rangle = \sum_i \lambda_i \langle\psi|\phi_i\rangle$ *for any* $\lambda_i \in \mathbb{C}$ *(linearity).*

2.1.2 The density operator

Due to imperfections or system interactions, many quantum states are not fully known in the real world. For that, we need a description of these quantum states. The *density operator* is a way to represent a quantum system that cannot be described by a single quantum state $|\psi\rangle$. Instead, it is in one of several quantum states $|\psi_i\rangle$, each appearing with corresponding probability p_i with $\sum_i p_i = 1$. The set $\{p_i, |\psi_i\rangle\}$ is called an *ensemble of pure states* with the completely known pure states $|\psi_i\rangle$. The density operator, often also called *density matrix* ρ of a quantum system, is given as

$$\rho = \sum_i p_i |\psi_i\rangle\langle\psi_i|. \tag{2.3}$$

The density operator fulfills the following conditions:

1. **Trace condition**: $Tr(\rho) = \sum_i \rho_{ii} = 1$ with diagonal entries ρ_{ii} of matrix ρ,
2. **Positivity condition**: ρ is a positive semi-definite operator, i.e., $\langle v|\rho|v\rangle \geq 0$ for any vector $|v\rangle \in \mathcal{H}$, and
3. **Hermiticity:** $\rho = \rho^\dagger$ with the Hermitian operator $\rho^\dagger = \left(\rho^T\right)^* = \begin{pmatrix} \rho_0^* & \rho_2^* \\ \rho_1^* & \rho_3^* \end{pmatrix}$ for a density matrix $\rho = \begin{pmatrix} \rho_0 & \rho_1 \\ \rho_2 & \rho_3 \end{pmatrix}$.

ρ^T is the transpose of the matrix and ρ^* the complex conjugate. The proofs of the trace and the positivity condition are given in [NC10] in Theorem 2.5.

Any quantum state that can be completely described as a state vector can also be represented by its density matrix $\rho = |\psi\rangle\langle\psi|$. In this case, the ensemble only contains one state vector that appears with probability $p = 1$. In the following, we call such a state a *pure state*. The density matrix of a pure state always fulfills the criterion $Tr(\rho^2) = 1$. All other quantum states that contain a mixture of several pure states $|\psi_i\rangle$ in their ensemble are called *mixed states*. The density operator of a mixed state fulfills $Tr(\rho^2) < 1$. A d-dimensional state of the form $\rho = \frac{\mathbb{1}}{d}$ we call the *maximally mixed state*.

2.1.3 Quantum bits

The *qubit* (short for a quantum bit) is the quantum analog of a classical bit in information theory. In this thesis, we consider the qubit as the fundamental quantum system. For now, we stick to the qubit as a mathematical object. The physical realization of such systems will be discussed later when we consider applications. The qubit lives in the two-dimensional Hilbert space $\mathbb{C}^2$. Compared to a classical bit (that can take values of either 0 or 1), a qubit can take any linear combination of states, the so-called *superposition*. Any state vector of a pure quantum state can be written in the following form:

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.4}$$

with complex numbers α and β. When we measure a qubit, we get more restricted results, meaning that we find the qubit in state $|0\rangle$ with probability $|\langle 0|\psi\rangle|^2 = |\alpha^2|$ and state $|1\rangle$ with probability $|\langle 1|\psi\rangle|^2 = |\beta^2|$. Since the probabilities must sum to one, i.e., $|\alpha|^2 + |\beta^2| = 1$, we find $|\psi\rangle$ to be normalized. The states $|0\rangle = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$ and $|1\rangle = \begin{pmatrix} 0 \\ 1 \end{pmatrix}$, that the qubit can take, form the so-called *computational basis*, i.e., these state vectors form an orthonormal basis of the vector space, here the two-dimensional Hilbert space $\mathcal{H}$.

Another useful representation of qubits is their geometric picture. Due to the

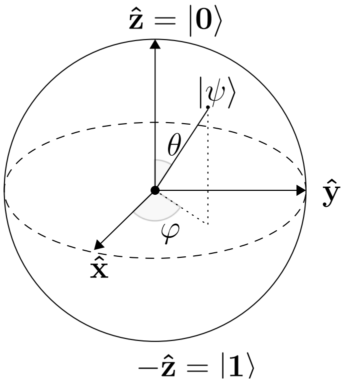

$\hat{\mathbf{z}} = |\mathbf{0}\rangle$
$|\psi\rangle$
θ
$\hat{\mathbf{y}}$
φ
$\hat{\mathbf{x}}$
$-\hat{\mathbf{z}} = |\mathbf{1}\rangle$

Figure 2.1: Bloch sphere: geometric representation of a qubit.

normalization property of a qubit, we can rewrite Eq. 2.4 in the following way:

$$|\psi\rangle = e^{i\gamma}\left(\cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle\right), \tag{2.5}$$

with real numbers γ, θ, and φ. Since the prefactor $e^{i\gamma}$ does not make a difference when observing a qubit (see Chapter 2 in [NC10] for more details), we can ignore it and effectively describe the qubit as

$$|\psi\rangle = \cos\frac{\theta}{2}|0\rangle + e^{i\varphi}\sin\frac{\theta}{2}|1\rangle. \tag{2.6}$$

This equation represents a sphere with θ and φ giving a specific point on the sphere. Fig. 2.1 shows the geometric representation of a qubit, often called the Bloch sphere. Pure quantum states are represented by any point on the surface of the Bloch sphere, while mixed quantum states lie in the interior of the Bloch sphere.

2.1.4 Composed quantum systems

In many applications, more than one quantum system will be of interest. That means that two or more qubits interact with each other or form a larger quantum system in a higher dimensional Hilbert space. The second postulate of quantum mechanics tells us how to describe such composed quantum systems.

Postulate 2.([NC10]) The state space of a composite physical system is the tensor product of the state space of the component physical systems. Moreover, if we have systems numbered 1 through n, and system number i is prepared in the state $|\psi\rangle_i$, then the joint state of the total system is $|\psi_1\rangle \otimes |\psi_2\rangle \otimes \cdots \otimes |\psi_n\rangle$.

Note that each state vector $|\psi_i\rangle$ lives in its own Hilbert space $\mathcal{H}_i$. The composed quantum system of n Hilbert spaces with equal dimensionality forms itself a Hilbert space, i.e., $\mathcal{H}_{1\ldots n} = \mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n$. In the following, the tensor product $|\psi_1\rangle \otimes |\psi_2\rangle$ is sometimes also written as $|\psi_1\rangle |\psi_2\rangle$ or $|\psi_1\psi_2\rangle$. Often, we use letters to indicate the affiliation of a (sub-)system to a party, i.e., $|\psi_A\psi_B \ldots\rangle$ with $|\psi_A\rangle \in \mathcal{H}_A, |\psi_B\rangle \in \mathcal{H}_B, \ldots$.

With the *reduced density operator*, it is possible to describe subsystems of a decomposed quantum system. Given a composed physical system of A and B described by the density operator ρ_{AB}, the reduced density operator to describe system A is given as

$$\rho_A = tr_B\left(\rho_{AB}\right). \tag{2.7}$$

The map of operators tr_B is called the *partial trace* over system B. It is defined as

$$tr_B\left(|a_1\rangle\langle a_2| \otimes |b_1\rangle\langle b_2|\right) \equiv |a_1\rangle\langle a_2|\, tr\left(|b_1\rangle\langle b_2|\right), \tag{2.8}$$

with vectors $|a_1\rangle, |a_2\rangle$ in Alice's state space and vectors $|b_1\rangle, |b_2\rangle$ in Bob's state space. For the trace on the right-hand side, we find $tr\left(|b_1\rangle\langle b_2|\right) = \langle b_2|b_1\rangle$. Given, for example, the state $\rho_{AB} = 1/2\left(|00\rangle + |11\rangle\right)_A \otimes \left(\langle 00| + \langle 11|\right)_B$, we find

$$\begin{aligned}
\rho_A &= tr_B(\rho_{AB}) \\
&= tr_B\left(\frac{|00\rangle\langle 00| + |00\rangle\langle 11| + |11\rangle\langle 00| + |11\rangle\langle 11|}{2}\right) \\
&= \frac{|0\rangle\langle 0|\langle 0|0\rangle + |0\rangle\langle 1|\langle 1|0\rangle + |1\rangle\langle 0|\langle 0|1\rangle + |1\rangle\langle 1|\langle 1|1\rangle}{2} \\
&= \frac{|0\rangle\langle 0| + |1\rangle\langle 1|}{2} \qquad (2.9) \\
&= \frac{\mathbb{1}}{2}. \qquad (2.10)
\end{aligned}$$

It is worth noticing that the reduced state is a mixed state, even though the joint system is pure and, therefore, fully describable.

2.1.5 Time evolution

So far, we have considered isolated quantum systems that do not interact with other systems or their environment. In real-world applications, it is not reasonable to make such an assumption. Therefore, we introduce the third postulate of quantum mechanics. This postulate deals with the time evolution of a quantum system that interacts with its environment.

Postulate 3.([NC10]) The evolution of a *closed* quantum system is described by a *unitary transformation.* That is, the state $|\psi\rangle$ of the system at time t_1 is related to the state $|\psi'\rangle$ of the system at time t_2 by a unitary operator U which depends only on the times t_1 and t_2,

$$|\psi'\rangle = U|\psi\rangle. \tag{2.11}$$

For any unitary operator U it holds $U^\dagger U = \mathbb{1}$, where $U^\dagger = (U^T)^*$ is the self adjoint of U. An operator is unitary if and only if all its matrix representations are unitary. The unitary operator U is also *normal*, which means that it holds $UU^\dagger = U^\dagger U$ from which $UU^\dagger = \mathbb{1}$ follows. Considering two state vectors $|\psi\rangle$ and $|\phi\rangle$ and a unitary operator U, such that $|\psi'\rangle = U\,|\psi\rangle$ and $|\phi'\rangle = U\,|\phi\rangle$. With $\langle\psi'| = \langle\psi|\,U^\dagger$, it holds

$$\langle\psi'|\phi'\rangle = \langle\psi|U^\dagger U|\phi\rangle = \langle\psi|\mathbb{1}|\phi\rangle = \langle\psi|\phi\rangle. \tag{2.12}$$

This shows that the unitary operator preserves the inner product.

2.1.6 Measurements

Every measurement performed on a quantum system influences the system. It is an interaction from the outside with the closed system. With a measurement, the experimentalist observes what is going on inside the quantum system. Therefore, the system is no longer closed and isolated from its surroundings. Before a measurement, the quantum system can be in a superposition of states. With a certain probability, one of these states is measured, projecting the quantum state onto one of its possible states. Postulate 4 describes the effects of a measurement performed on a quantum system.

Postulate 4.([NC10]) Quantum measurements are described by a collection $\{M_m\}$ of *measurement operators*. These are operators acting on the state space of the system being measured. The index m refers to the measurement outcome that may occur in the experiment. If the state of the quantum system is $|\psi\rangle$ immediately before the measurement, then the probability that result m occurs is given by

$$p(m) = \langle\psi|M_m^\dagger M_m|\psi\rangle, \tag{2.13}$$

and the state of the system after the measurement is

$$\frac{M_m|\psi\rangle}{\sqrt{\langle\psi|M_m^\dagger M_m|\psi\rangle}}. \tag{2.14}$$

Note that the measurement operators M_m fulfill completeness:

$$\sum_m M_m^\dagger M_m = \mathbb{1} \tag{2.15}$$

and probabilities sum to one:

$$1 = \sum_m p(m) = \sum_m \langle\psi|M_m^\dagger M_m|\psi\rangle. \tag{2.16}$$

The postulate tells us that each measurement leads to one measurement outcome m with a certain probability $p(m)$. After the measurement, the quantum state changes due to the interaction. The new state is given by Eq. 2.14. A simple but important example we use in our work is the measurement of a single qubit in the computational basis. The two measurement operators $M_0 = |0\rangle\langle 0|$ and $M_1 = |1\rangle\langle 1|$ give the measurement with possible outcomes 0 or 1. The completeness relation from Eq. 2.15 is fulfilled since $M_0^\dagger M_0 + M_1^\dagger M_1 = M_0 + M_1 = \mathbb{1}$. Given a single qubit state by

$$|\psi\rangle = \alpha|0\rangle + \beta|1\rangle, \tag{2.17}$$

we obtain the measurement outcome 0 with probability

$$p(0) = \langle\psi|M_0^\dagger M_0|\psi\rangle = \langle\psi|M_0|\psi\rangle = |\alpha|^2, \tag{2.18}$$

and analogously, for a measurement outcome 1, we obtain a probability of $|\beta|^2$. The states we obtain after the measurement are given by:

$$\frac{M_0|\psi\rangle}{|\alpha|} = \frac{\alpha}{|\alpha|}|0\rangle \tag{2.19}$$

$$\frac{M_1|\psi\rangle}{|\beta|} = \frac{\beta}{|\beta|}|1\rangle. \tag{2.20}$$

Effectively, the prefactors $\alpha/|\alpha|$ and $\beta/|\beta|$ can be neglected so that we obtain the state $|0\rangle$ when measuring the outcome 0, and we obtain the state $|1\rangle$ when measuring the outcome 1, respectively.

A special set of measurement operators form the projective measurements. This can be seen as a special case of the previous postulate in which the measurement operators $\{M_m\}$ additionally fulfill orthogonality: the operators are Hermitian and fulfill $M_m M_{m'} = \delta_{mm'} M_m$.

Projective measurements. A projective measurement is described by an *observable* M, a Hermitian operator on the state space of the system being observed. The observable has a spectral decomposition

$$M = \sum_m m P_m, \tag{2.21}$$

where P_m is the projector onto the eigenspace of M with eigenvalue m. The possible outcomes of the measurement correspond to the eigenvalues, m, of the observable. Upon measuring the state $|\psi\rangle$, the probability of getting result m is given by

$$p(m) = \langle\psi|P_m|\psi\rangle. \tag{2.22}$$

Given the outcome m occurred, the state of the quantum system immediately after the measurement is

$$\frac{P_m|\psi\rangle}{\sqrt{p(m)}}. \tag{2.23}$$

A measurement can also be performed on a composed quantum system. Consider, for example, the situation where Alice and Bob share a state ρ_{AB} with $\mathcal{H}_{AB} = \mathcal{H}_A \otimes \mathcal{H}_B$, and each party wants to perform a measurement given by the measurement operators $\{M_A\}$ and $\{M_B\}$, respectively. The measurement outcome of Alice performed on the reduced state $tr_B\rho_{AB}$ has to agree with the measurement $\{M_A \otimes \mathbb{1}_B\}$ performed on the composed quantum system ρ_{AB}. The same has to hold for Bob's measurement.

2.2 Entanglement

The property of entanglement is one of the fundamental concepts that arises only in composed quantum systems. Entanglement is one of the central advantages of quantum mechanics. In quantum communication systems, for example, entanglement can be used to generate secret keys between the communicating parties. We will discuss this in detail in Sec. 2.4.

Given an entangled composed quantum system of n subsystems with Hilbert space $\mathcal{H}_1 \otimes \mathcal{H}_2 \otimes \cdots \otimes \mathcal{H}_n$, it is impossible to describe each quantum system independently, i.e., no product of the subspaces can be generated to describe the composed system. The concept of entanglement is independent of the distance between the quantum systems forming an entangled system. Once a measurement is performed on an entangled system, all subsystems contained in the composed system are influenced by the measurement. Even without sharing the measurement result, the total composed system is influenced by the measurement outcome, so that the previous measurement now predetermines a measurement on any subsystem.

2.2.1 Bipartite entanglement

Given any bipartite system with Hilbert spaces $\mathcal{H}_{\mathcal{A}}$ and $\mathcal{H}_B$, the decomposed system $|\psi\rangle_{AB} \in \mathcal{H}_{\mathcal{AB}}$ is entangled if and only if

$$|\psi\rangle_{AB} \neq |\psi\rangle_A \otimes |\psi\rangle_B \tag{2.24}$$

i.e., the composed system is not describable by the product of its states $|\psi\rangle_A$ and $|\psi\rangle_B$. The well-known maximally entangled states in a bipartite setup are the four *Bell states* of the following form:

$$|\phi^{\pm}\rangle = \frac{1}{\sqrt{2}}\left(|00\rangle \pm |11\rangle\right) \tag{2.25}$$

$$|\psi^{\pm}\rangle = \frac{1}{\sqrt{2}}\left(|01\rangle \pm |10\rangle\right). \tag{2.26}$$

On the other hand, a state is separable if each subsystem of the composed system can be described independently of each other:

$$|\psi\rangle_{AB} = |\psi\rangle_A \otimes |\psi\rangle_B. \tag{2.27}$$

An example of a separable state is given by:

$$|\psi\rangle_{AB} = \frac{1}{\sqrt{2}}\left(|00\rangle + |01\rangle\right) = \frac{1}{\sqrt{2}}|0\rangle_A \left(|0\rangle + |1\rangle\right)_B. \tag{2.28}$$

Testing whether a given state is entangled or separable is not an easy task. There are different criteria that can be used to detect separability. For example, [GT09] gives an overview of different separability criteria for discrete-variable systems, and [FVMH19] deals with the techniques and challenges in experimental entanglement certification in such systems.

The authors in [VPRK97, DHR02], derive different quantities to quantify the amount of entanglement are given. In our work, we address the question of how close a given state is to a known entangled quantum state. To determine this, we use the *fidelity* F, which is a measure of distance. The fidelity gives the distance between two states ρ and σ as

$$F(\rho, \sigma) = \left(tr\sqrt{\rho^{1/2}\sigma\rho^{1/2}} \right)^2 . \tag{2.29}$$

In the case of considering the fidelity between a pure state $|\psi\rangle$ and any state ρ, the fidelity is given by the overlap between the state $|\psi\rangle$ and ρ

$$F(|\psi\rangle, \rho) = \langle\psi|\rho|\psi\rangle. \tag{2.30}$$

With this, we can calculate how close an arbitrary state ρ is to a Bell state $|\phi^{\pm}\rangle$ or $|\psi^{\pm}\rangle$.

Bell measurement

To create or measure a Bell state, we need a certain set of operations that we can apply to a quantum state. We can perform a *quantum computation*, the quantum analog to classical computation, to change a quantum state. Just like with classical computers, quantum computers are represented by *quantum circuits*. These consist of wires and elementary *quantum gates*. This allows the information to be manipulated and transmitted in the desired manner. In quantum circuits, we find single and multiple qubit gates. A single qubit gate gets one qubit as the input and operates on that single qubit. In classical computation, such a gate is the *NOT gate*. It changes a bit from 0 → 1 and vice versa (1 → 0). In quantum computing, all *Pauli matrices* are single qubit gates that act, in principle, on a superposition ($|\psi\rangle = \alpha|0\rangle + \beta|1\rangle$). The Pauli X-operator acts similarly to the classical NOT gate. It takes a superposition and changes the role of the states $|0\rangle$ and $|1\rangle$:

$$\begin{aligned} X|\psi\rangle &= X(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|1\rangle + \beta|0\rangle. \end{aligned} \tag{2.31}$$

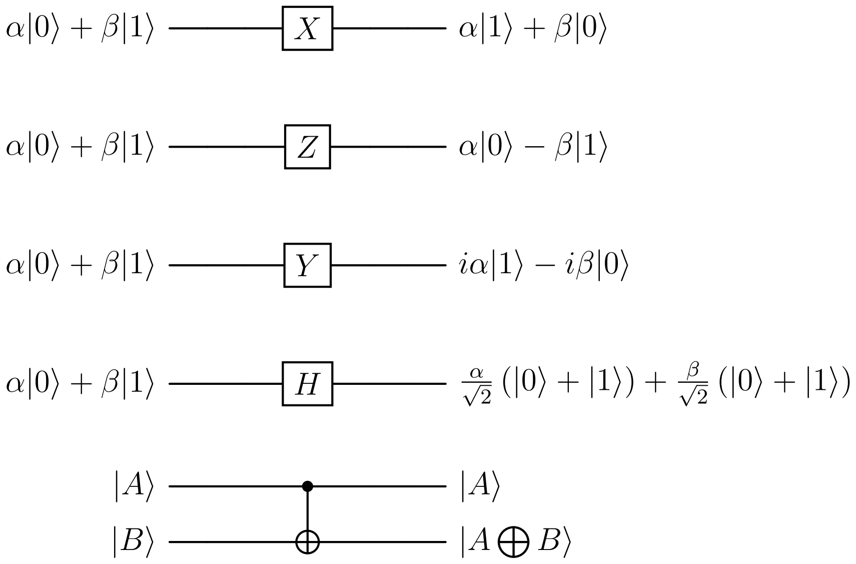

$\alpha|0\rangle + \beta|1\rangle$
X
$\alpha|1\rangle + \beta|0\rangle$
$\alpha|0\rangle + \beta|1\rangle$
Z
$\alpha|0\rangle - \beta|1\rangle$
$\alpha|0\rangle + \beta|1\rangle$
Y
$i\alpha|1\rangle - i\beta|0\rangle$
$\alpha|0\rangle + \beta|1\rangle$
H
$\frac{\alpha}{\sqrt{2}}(|0\rangle + |1\rangle) + \frac{\beta}{\sqrt{2}}(|0\rangle + |1\rangle)$
$|A\rangle$
$|B\rangle$
$|A\rangle$
$|A \bigoplus B\rangle$

Figure 2.2: Circuit representation of the three Pauli matrices X, Z, and Y, the Hadamard gate H, and the two-qubit CNOT-gate. The $\bigoplus$ is the sum modulo two operation.

The Pauli X-operator is represented by the matrix

$$X = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix}. \tag{2.32}$$

The Pauli Z-operator changes the phase between the state of a superposition, i.e., it changes the sign of the state $|1\rangle$ to $-|1\rangle$:

$$\begin{aligned} Z|\psi\rangle &= Z(\alpha|0\rangle + \beta|1\rangle) \\ &= \alpha|0\rangle - \beta|1\rangle. \end{aligned} \tag{2.33}$$

with Pauli Z-operator

$$Z = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}. \tag{2.34}$$

Combining both the Pauli X- and the Pauli Z-operator gives the third Pauli operator, which is called Pauli Y-operator

$$Y = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix} . \tag{2.35}$$

Another useful single qubit gate we will need to perform Bell state measurements is the *Hadamard gate*. This gate takes a single quantum state and returns a superposition:

$$H|0\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle + |1\rangle\right) \tag{2.36}$$

$$H|1\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle - |1\rangle\right) . \tag{2.37}$$

The Hadamard gate is given by the matrix

$$H = \frac{1}{\sqrt{2}} \begin{pmatrix} 1 & 1 \\ 1 & -1 \end{pmatrix} . \tag{2.38}$$

With a multiple qubit gate, multiple qubits are received as input. Unlike classical gates, all input qubits are returned as outputs. The most important multiple qubit gate for this work is the so-called *controlled-NOT gate*, or *CNOT gate* for short. This gate gets two qubits as input: one *control qubit* and one *target qubit*. Based on the state of the control qubit, the target qubit is either flipped or remains untouched. Its action is given by the matrix

$$U_{CN} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{pmatrix} . \tag{2.39}$$

The truth table of the CNOT-gate shows the changes it performs on the two input qubits:

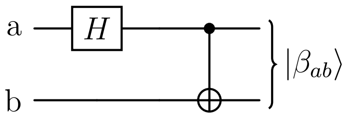


Figure 2.3: Quantum circuit for the Bell state creation.

Input		Output	
Control	Target	Control	Target
$\|0\rangle$	$\|0\rangle$	$\|0\rangle$	$\|0\rangle$
$\|0\rangle$	$\|1\rangle$	$\|0\rangle$	$\|1\rangle$
$\|1\rangle$	$\|0\rangle$	$\|1\rangle$	$\|1\rangle$
$\|1\rangle$	$\|1\rangle$	$\|1\rangle$	$\|0\rangle$

Fig. 2.2 gives an overview of the circuit representation of the quantum gates introduced here.

To create a Bell state, the quantum circuit from Fig. 2.3 is performed. Depending on the two input qubits, here named a and b, one of the four Bell states from Eq. 2.25 or Eq. 2.26 respectively is created:

In	Out
$\|00\rangle$	$\frac{1}{\sqrt{2}}\left(\|00\rangle + \|11\rangle\right)$
$\|01\rangle$	$\frac{1}{\sqrt{2}}\left(\|01\rangle + \|10\rangle\right)$
$\|10\rangle$	$\frac{1}{\sqrt{2}}\left(\|00\rangle - \|11\rangle\right)$
$\|11\rangle$	$\frac{1}{\sqrt{2}}\left(\|01\rangle - \|10\rangle\right)$

Starting for example with the input state $|00\rangle$, the quantum circuit leads to the following:

$$|00\rangle_{AB} \xrightarrow{H_A \otimes \mathbb{1}_B} \frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)_A |0\rangle_B \tag{2.40}$$

$$\frac{1}{\sqrt{2}}\left(|0\rangle + |1\rangle\right)_A |0\rangle_B \xrightarrow{U_{CN}} \frac{1}{\sqrt{2}}\left(|00\rangle + |11\rangle\right). \tag{2.41}$$

Due to the Hadamard gate, the first qubit is converted to a superposition. Note that the second qubit remains untouched, i.e., the identity is applied to qubit b.

This is followed by the CNOT gate. The superposition created before is given as the control qubit. The gate flips the state of qubit b if a is set to 1.

Entanglement swapping

We now introduce the concept of *entanglement swapping*. We will use this operation later in the context of quantum networks. In many applications, such as quantum networks, entanglement is the basic resource we need to transfer information. Therefore, it is necessary to distribute entangled quantum states between the communicating parties. Entanglement swapping is used to make the communication feasible even over larger distances. With this concept, entanglement can be interchanged between distant parties that have never interacted directly with each other. Entangled states are created locally by the parties, here, A and B. In the next step, both parties keep one qubit of the entangled pair locally while the second qubit is sent to a central station. A *Bell measurement* is performed at this station between arriving qubits. This causes the remaining qubits to be projected onto one of the four Bell states. The Bell measurement uses the same quantum gates as the quantum circuit to create a Bell state. However, the sequence in which the gates are performed for the Bell state measurement is inverted. Fig. 2.4 shows an exemplary setup for the entanglement swapping. The protocol performed by the parties is as follows:

1. Alice and Bob both create the entangled state $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$,

2. Both parties hold one qubit locally (say A_1 and B_1) and send the second qubit to the central station (say A_2 and B_2),

3. A Bell state measurement is performed at the arrived qubits:

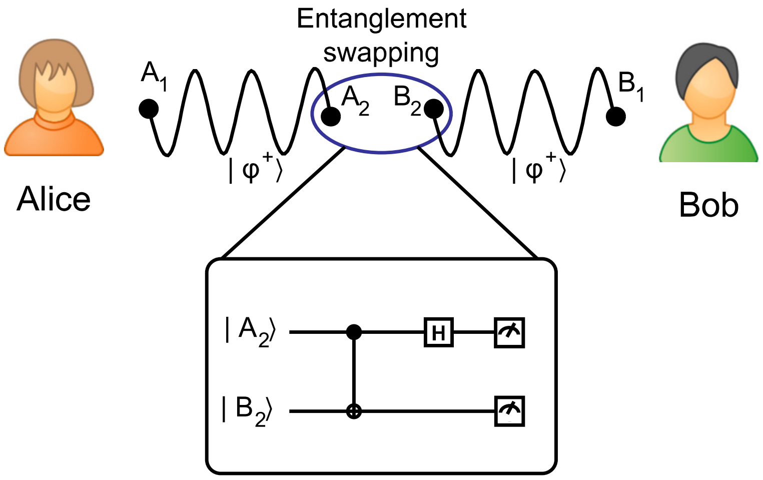


Figure 2.4: Scheme of the entangement swapping setup performed between two parties, Alice and Bob. The last symbol within the quantum circuit ( ) represents the measurement performed on the qubits A_2 and B_2.

a) The input is given by

$$\begin{aligned} \left|\phi^+\right\rangle_A \otimes \left|\phi^+\right\rangle_B &= \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right)_A \otimes \frac{1}{\sqrt{2}} \left(|00\rangle + |11\rangle\right)_B \qquad (2.42) \\ &= \frac{1}{2} \big(|0000\rangle_{A_1A_2B_1B_2} + |0011\rangle_{A_1A_2B_1B_2} \\ &+ |1100\rangle_{A_1A_2B_1B_2} + |1111\rangle_{A_1A_2B_1B_2} \big) \qquad (2.43) \\ &= \frac{1}{2} \big(|00\rangle_{A_1B_1} |00\rangle_{A_2B_2} + |01\rangle_{A_1B_1} |01\rangle_{A_2B_2} \\ &+ |10\rangle_{A_1B_1} |10\rangle_{A_2B_2} + |11\rangle_{A_1B_1} |11\rangle_{A_2B_2} \big) \,. \qquad (2.44) \end{aligned}$$

Note that the order of the qubits in the last step is changed such that the qubits held locally by the parties and those sent to the central station are each grouped together.

b) First, the CNOT gate is performed between qubit A_2 (control qubit) and

B_2 (target qubit) leading to the following:

$$\frac{1}{2}\left(\left|00\right\rangle_{A_1B_1}\left|00\right\rangle_{A_2B_2}+\left|01\right\rangle_{A_1B_1}\left|01\right\rangle_{A_2B_2}+\left|10\right\rangle_{A_1B_1}\left|11\right\rangle_{A_2B_2}\right.$$
$$\left.+\left|11\right\rangle_{A_1B_1}\left|10\right\rangle_{A_2B_2}.\right. \tag{2.45}$$

c) Applying the Hadamard gate on qubit A_2 and rearranging the terms gives:

$$\frac{1}{2}\left(\frac{1}{\sqrt{2}}\left|00\right\rangle_{A_1B_1}(\left|0\right\rangle+\left|1\right\rangle)_{A_2}\left|0\right\rangle_{B_2}+\frac{1}{\sqrt{2}}\left|01\right\rangle_{A_1B_1}(\left|0\right\rangle+\left|1\right\rangle)_{A_2}\left|1\right\rangle_{B_2}\right.$$
$$\left.+\frac{1}{\sqrt{2}}\left|10\right\rangle_{A_1B_1}(\left|0\right\rangle-\left|1\right\rangle)_{A_2}\left|1\right\rangle_{B_2}+\frac{1}{\sqrt{2}}\left|11\right\rangle_{A_1B_1}(\left|0\right\rangle-\left|1\right\rangle)_{A_2}\left|0\right\rangle_{B_2}\right) \tag{2.46}$$
$$=\frac{1}{2}\left(\frac{1}{\sqrt{2}}(\left|00\right\rangle+\left|11\right\rangle)_{A_1B_1}\left|00\right\rangle_{A_2B_2}+\frac{1}{\sqrt{2}}(\left|00\right\rangle-\left|11\right\rangle)_{A_1B_1}\left|10\right\rangle_{A_2B_2}\right.$$
$$\left.+\frac{1}{\sqrt{2}}(\left|01\right\rangle+\left|10\right\rangle)_{A_1B_1}\left|01\right\rangle_{A_2B_2}+\frac{1}{\sqrt{2}}(\left|01\right\rangle-\left|10\right\rangle)_{A_1B_1}\left|11\right\rangle_{A_2B_2}\right) \tag{2.47}$$
$$=\frac{1}{2}\left(\left|\phi^+\right\rangle_{A_1B_1}\left|00\right\rangle_{A_2B_2}+\left|\phi^-\right\rangle_{A_1B_1}\left|10\right\rangle_{A_2B_2}\right.$$
$$\left.+\left|\psi^+\right\rangle_{A_1B_1}\left|01\right\rangle_{A_2B_2}+\left|\psi^-\right\rangle_{A_1B_1}\left|11\right\rangle_{A_2B_2}\right). \tag{2.48}$$

d) The central station performs a measurement in the computational basis on the qubits A_2 and B_2 that are stored in the memories. Each measurement outcome $\{\left|00\right\rangle,\left|01\right\rangle,\left|10\right\rangle,\left|11\right\rangle\}$ of system A_2B_2 occurs with probability $p=|\frac{1}{2}|^2=\frac{1}{4}$. Depending on the measurement outcome, the subsystem of qubits held locally by the parties (i.e., A_1B_1) is then in one of the four Bell states $\{\left|\phi^+\right\rangle,\left|\phi\right\rangle,\left|\psi^+\right\rangle,\left|\psi^-\right\rangle$. Mathematically, this is obtained by tracing out system A_2B_2 (i.e., $\rho_{A_1B_1}=tr_{A_2B_2}(\rho_{A_1B_1A_2B_2})$). Note that the created Bell state can be changed to a specific Bell state by performing local measurements on subsystems A_1 and B_1, depending on the measurement outcome of the system A_2B_2. To get the Bell state $\left|\phi^+\right\rangle$ from the actual Bell state the parties hold locally, the following local operations need to be performed:

$$Z^{a_2}X^{b_2}, \tag{2.49}$$

with $a_2, b_2 \in \{0,1\}$ being the measurement results of the qubits A_2 and B_2.

For example, if the parties desire to share the $|\phi^+\rangle$ state, but the measurement in the central station leads to the outcome $|10\rangle$, the parties must perform a Z measurement locally since $a_2 = 1$ and $b_2 = 0$ so that Z^1X^0. It is similarly possible to create any other Bell state. In that case, the local operations performed based on the measurement outcome must be adjusted accordingly.

2.2.2 Genuine multipartite entanglement

So far, we have focused on bipartite systems of two qubits. Entanglement also occurs in larger systems in which many qubits interact with each other. The straightforward way to define entanglement in a multipartite system is via the concept of *full entanglement* or *full separability*, respectively. A multipartite quantum system of n qubits $|\psi\rangle_{12\ldots n} \in \mathcal{H}_{12\ldots n}$ each with Hilbert space $\mathcal{H}_i$ with $i \in \{1,2,\ldots,n\}$ is *fully separable*, if and only if

$$|\psi\rangle = \otimes_{i=1}^{n} |\psi\rangle_i\,. \tag{2.50}$$

Accordingly, a system is *fully entangled* if and only if all bipartite partitions are pairwise entangled. This is called *genuine multipartite entanglement* (GME).

Unlike in bipartite entanglement, there are more classifications of entanglement in the multipartite setup. Besides the fully entangled or fully separable states, there are classes of partially separable states. This means that a given state of n subsystems is separable with respect to a partition of all subsystems. A detailed overview of the concepts of multipartite entanglement is given, for example, in [HHHH09].

In our work, however, we focus on the class of genuine multipartite entangled states. The well-known state we use here is the so-called *Greenberger-Horne-Zeilinger state* (GHZ state) [GHZ89]. The GHZ state is a fully entangled quantum state of at least three qubits. The general equation describing any GHZ state with n qubits

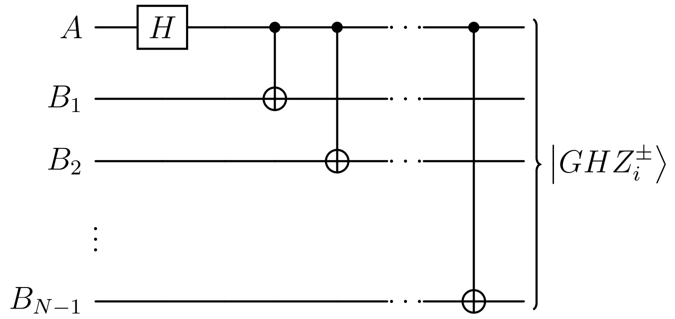


Figure 2.5: Quantum circuit for the GHZ state creation. When passing through the circuit from right to left, the GHZ state measurement that is used in entanglement swapping is given.

is given by:

$$\left|GHZ_i^{\pm}\right\rangle = \frac{1}{\sqrt{2}}\left(|i\rangle \pm \left|n^2 - 1 - i\right\rangle\right) \quad \text{with } \; i = 0, 1, \ldots, 2^{n-1} - 1. \tag{2.51}$$

Here, the numbers within the kets are written in binary notation. Often, this particular state is considered:

$$|GHZ\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}\right) \text{ for } n \geq 3. \tag{2.52}$$

This is the generalization of the $|\phi^+\rangle$ Bell state for n qubits.

GHZ state creation and measurement

The creation and distribution of the GHZ state via entanglement swapping work analogously to the Bell state circuit presented in Sec. 2.2.1. Fig. 2.5 shows the corresponding quantum circuit for the N-partite setup with N parties. We can create a GHZ state by moving left to right. In contrast, a GHZ state measurement is performed by running through the circuit from right to left, starting with the CNOT gates. This is used in entanglement swapping when not just two but N parties want to distribute a genuine multipartite entangled state (here the GHZ state) among themselves. Analogously to before, each party prepares a Bell pair (say the $|\phi^+\rangle$ state) and sends one qubit to the central station while keeping the second one locally. Due to the measurement performed at the central station, the qubits held locally are projected onto a GHZ state. With classical communication of the measurement results (of the qubit at the central station) and local operations (LOCC), we can generate the desired GHZ state. Again, the subsystem of qubits held locally is obtained by tracing out the central station and the qubits stored in it.

Due to the structure of the quantum circuit, one party has a special role (namely, performing the Hadamard gate and providing the control qubit for all the CNOT gate operations). This party we call the *center party* A, and the other parties we call the *peers* B_i with $i \in \{1, 2, \ldots, N-1\}$.

2.3 Quantum networks

Quantum networks – an important element of quantum computing – consist of end nodes with quantum processors and channels that connect these processors. Similar to classical networks, information (in the form of qubits) is processed and transmitted in these networks. Quantum processors perform quantum circuits on the qubits to solve certain tasks. A quantum network can form a cluster of many quantum processors to increase the computational power of the combination of all processors. This is used in quantum computing to get more powerful computers. Quantum networks can also be used for communication. For this purpose, information can be shared and transmitted between the end nodes via the channels. Entangled states between the nodes are often required to send information or share secret keys. Therefore, entanglement forms a basic resource needed in quantum networks.

Besides the end nodes and the channels connecting the quantum processors, intermediate stations for long-distance communication are an essential component of quantum networks. These *quantum repeaters* are introduced in the following subsection. The distribution of entanglement in quantum networks via quantum repeaters will be the main topic of the following sections and our work.

2.3.1 Quantum repeater

Quantum repeaters play an essential role in quantum communication over longer distances. Due to the interaction of the qubits with the fiber or its environment, the distance over which qubits can be sent is limited. In the following, we consider qubits as a physical system of depolarized photons. The information is then encoded as follows: the state $|0\rangle$ is given by a photon with horizontal polarization, and the state $|1\rangle$ is encoded as the photon with vertical polarization. The superposition is given by any other polarization that leads – when being measured – to either the horizontal or vertical polarization, each with a certain probability. As a channel, we consider optical fibers such as the standard telecom fibers. In this case, the distance d that a qubit can travel through the fiber, and the probability p that the qubit arrives successfully are related via:

$$p = 10^{-\frac{\alpha d}{10}}. \tag{2.53}$$

With an absorption coefficient $\alpha = 0.2$ dB/km [1], we find that qubits arrive successfully after a distance $d = 100$ km only with a probability of $p = 0.01$. So only 1% of all qubits sent over a fiber of 100 km arrive successfully. Under these circumstances, building a quantum network would be very challenging. Either one would have to have a quantum processor every few kilometers, or one would have to send a lot of qubits to compensate for the high loss rate of the fiber over the distance. In practice, intermediate stations are often used. However, these stations are not end nodes equipped with full quantum processors. Instead, quantum repeaters are placed between the end nodes to enlarge the possible communication distance between them.

In classical networks, amplifiers are used to increase the signal during its transition in a fiber. With quantum information, using classical amplifiers is infeasible due to the so-called *no-cloning theorem* [WZ82]. The theorem states that doing an exact copy of an unknown quantum state is impossible. Therefore, the use of amplifiers for quantum information would require a predetermination of the complete quantum state sent through the channel. Quantum repeaters are used to solve this problem. They are elements that are placed between two end nodes, performing a Bell state measurement on the arrived qubits. Based on entanglement swapping, quantum repeaters work differently from classical amplifiers. The protocol for the

[1] The value $\alpha = 0.2$ dB/km belongs to a commercial optical fiber that is used with light of the wavelength of $\lambda = 1550$ nm.

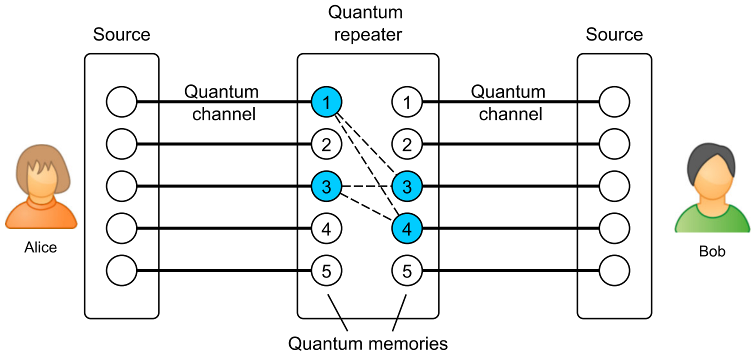


Figure 2.6: Representation of a bipartite quantum repeater with multiplexing (adapted from [AKB14a]). Each party has five memories and sources available, so up to five Bell state measurements can be performed in parallel. Light blue quantum memories indicate that they are filled with a qubit. Possible connections for the entanglement swapping are indicated with the dotted lines.

entanglement swapping is the same as in Sec. 2.2.1. Both parties send one qubit of an entangled state to the quantum repeater. The repeater station performs a Bell state measurement to project the locally held qubits on one of the four Bell states. By performing additional local operations, the desired Bell state can be preserved. The precise protocol about the working principle of a quantum repeater is given in the following subsection on *multiplexing*. How the entangled states are used for communication or secret key distribution will be discussed in Sec. 2.4.

Multiplexing

Collins et al. first introduced *multiplexing* to a quantum repeater in 2007 [CJKK07]. Multiplexing means every party has several quantum memories available in the quantum repeater. We denote the number of memories per party by m. In addition to the quantum memories, each party has m quantum sources, each connected via a quantum channel to one of the memories. Fig. 2.6 shows an example of a quantum repeater with $m = 5$ memories per party. Due to parallel channels between the end node and the quantum repeater, up to m Bell pairs can be generated in parallel. Therefore, up to m Bell state measurements are performed simultaneously. Of course, this depends on the number of qubits available at the quantum repeater. The goal of the multiplexing setup is to maximize the number of Bell pairs that can be distributed in each protocol round between the end nodes. One round of the protocol consists of the following steps:

1. Each party prepares a Bell pair and sends one qubit to the quantum router while the second qubit is held locally. Note that this is done only for empty memories since filled memories are not overwritten with new qubits.

2. Qubits that arrive successfully at the quantum repeater are stored in the memories. This event is heralded by the central station.

3. The quantum repeater performs Bell state measurements (compare the protocol steps for the entanglement swapping in Sec. 2.2.1) on a maximal number of qubits in parallel. This is based on the bipartite *matching problem*, which is well-known in graph theory (see [LP09], for example). The matching problem defines how the quantum memories are connected for the Bell state measurements.

4. The measurement outcome is heralded. The parties can perform additional local operations to turn the entangled state into the desired Bell state.

5. All quantum memories used for a measurement are emptied again. All other memories remain untouched. The actual memory configuration of filled and empty memories is the starting configuration of the following round.

Fig. 2.7 gives an overview of the quantum memories in the repeater station during one protocol round. The multiplexing focuses on step 3 of the protocol. In this step, the advantage of performing up to m Bell state measurements in parallel comes into account. The way the underlying matching problem works is explained in the following.

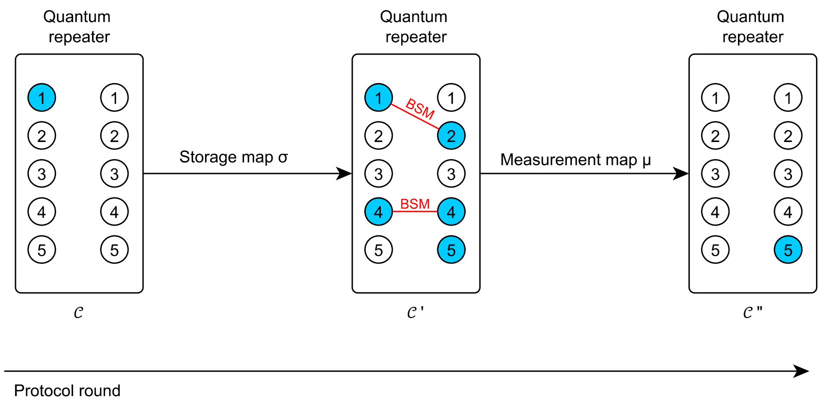


Figure 2.7: Illustration of a protocol round. An example configuration of the quantum memories within the quantum repeater during a round is shown. Filled memories are depicted in blue.

Bipartite matching

In each protocol round, the matching is performed individually on the memory configuration given in the quantum repeater. The quantum repeater can be seen as a *bipartite graph* $G = (V, E)$ with nodes V and edges E. The quantum memories are the graph's nodes, and the possible connections between the quantum memories (nodes) are represented by the edges. Due to the bipartition of the memories, the graph is bipartite: quantum memories belong either to party A or to party B, and edges are only drawn between nodes from different subsets V_A or V_B, i.e., $E = \{\{v_i, v_j\} | v_i \in V_A \wedge v_j \in V_B\}$. The bipartite graph looks different every round, depending on the memories filled in that round. Note that only filled memories contribute to the graph. The *matching* M outputs an independent set of edges that do not share common nodes. This property is required, as no qubit can appear in two different Bell state measurements. Since we want to maximize the number of Bell state measurements per round, we want that edge set to be maximal. A matching is chosen as follows:

- The bipartite graph is created from the given quantum repeater setup every round: all filled memories contribute to the graph, with the allocation to the subset V_A or V_B given by its allocation to party A or B. All edges indicating the possible connections between the nodes are drawn. The edges must fulfill the following properties:

  1. Edges always consist of exactly two nodes, one from each subset.
  2. The *connection length* w defines which nodes are allowed to be connected (see Fig. 2.8(a) for an example). The connection length is given by the difference in the labels of the quantum memories being connected. For example, $w = 0$ means that only memories with the same label can be connected, whereas $w = m - 1$ allows all connections between the memories independent of their label.

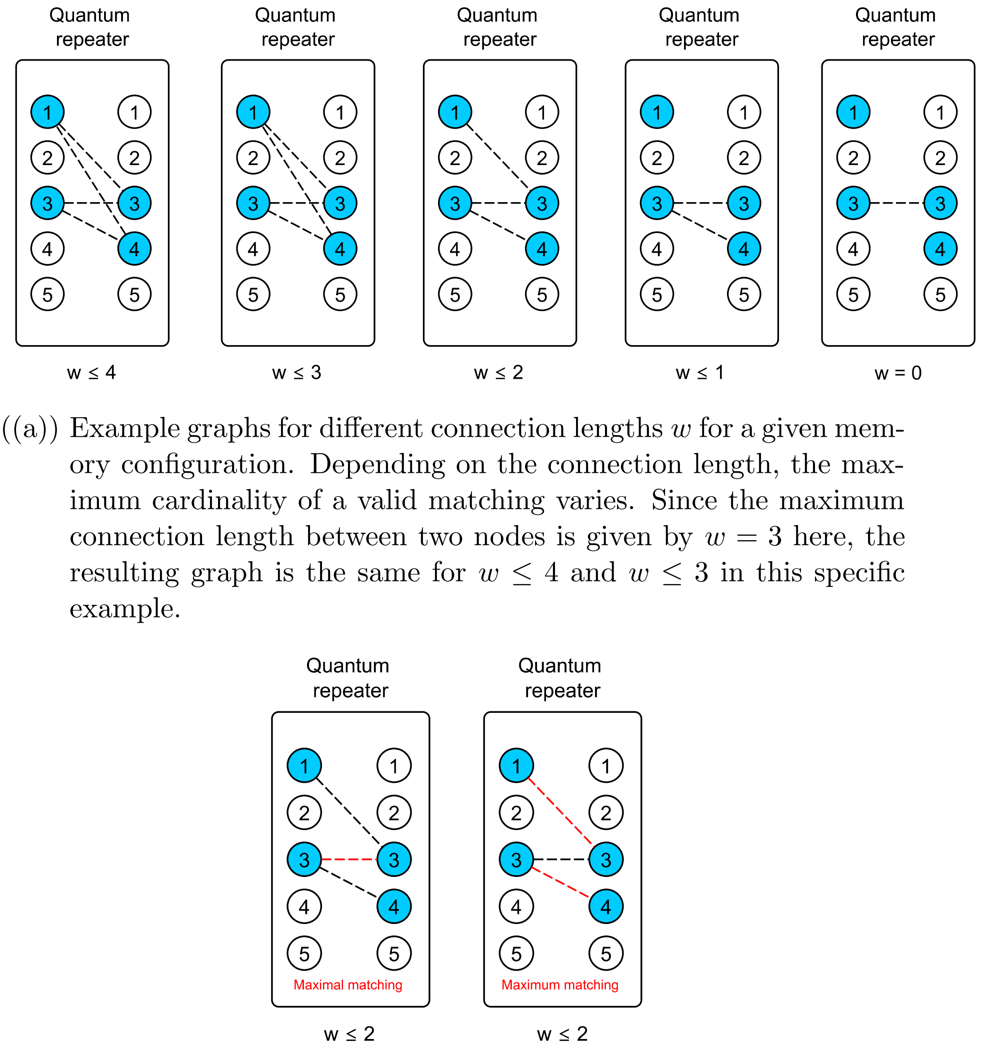
- Given the set of all possible edges fulfilling the properties from above, an edge set with maximum cardinality and no two common vertices is chosen. An example is given in Fig. 2.8(b).

The connection length is taken into account due to the assumption that it can be infeasible or at least experimentally challenging to realize Bell state measurements between all memories from the parties.

The *maximum cardinality bipartite matching* is well-known in graph theory and can be solved in polynomial time, for example, with the algorithm by Hopcroft and Karp (1973) [HK71]. For the implementation of the bipartite matching, complete packages, for example, exist for Python. In this work, we implement the bipartite matching with the package *bipartite* from the *networkx* library [Net24a].

Repeater rate

Due to multiplexing, more than one Bell state measurement can be performed in each protocol round. The aim is to maximize the number of entangled states distributed in every round. The figure of merit that describes how many Bell state measurements ℓ are successfully performed on average per memory and round is the *repeater rate*. The average number of successful Bell state measurements is denoted as $\langle \ell \rangle$. We



((a)) Example graphs for different connection lengths w for a given memory configuration. Depending on the connection length, the maximum cardinality of a valid matching varies. Since the maximum connection length between two nodes is given by $w = 3$ here, the resulting graph is the same for $w \leq 4$ and $w \leq 3$ in this specific example.

((b)) Example matching (shown in red) for the graph with connection length $w \leq 2$. The matching on the left shows a maximal matching. This is a matching to which no additional edges can be added without obtaining duplicate nodes. The matching on the right is a maximum matching with maximum cardinality. This matching would be valid for the maximum cardinality bipartite matching applied to the quantum repeater.

Figure 2.8: Example graphs for a specific memory configuration in the quantum repeater. The different graphs resulting from restrictions on the connection length are given in (a). Fig. (b) shows two different matchings for the graph with a connection length of $w \leq 2$.

average the repeater rate over all running rounds up to the current round s_c:

$$R(s_c) = \frac{1}{s_c} \sum_{s=1}^{s_c} \frac{\langle \ell \rangle}{m}. \tag{2.54}$$

In [AKB14a], a detailed representation of how the repeater rate can be calculated analytically is given. Here, we give some notation and general ideas of the concepts that we will later need for the multipartite setup.

Mathematically, we can describe the quantum repeater with its changes regarding the filled memories as a *Markov chain* [MT09]. A Markov chain is a stochastic process that describes the sequence of possible *events*, each only depending on the state of the previous event. The configuration of the quantum memories within the quantum repeater can be given as a binary bit string $\mathcal{C} = \{0,1\}^{2m}$ of total length $2m$. The string can be divided into two substrings (each of length m): one representing Alice's memories and one for Bob's memories. For example, the configuration describing the quantum repeater in Fig. 2.8, is $\mathcal{C} = \{1010000110\}$. Filled memories are represented by a one, and empty memories by a zero. Now, we transfer that to the description of Markov chains. In the quantum repeater setup, the state of such an event describes the configuration of the quantum memories at one round of the protocol. The memory configuration at the beginning of a round is given by the final memory configuration of the previous round. The statistical process through which such a configuration was created does not matter and is independent of the further course of the Markov chain. The *state vector* π is a column vector of length 2^{2m} that gives the probabilities to find the memories within the quantum router in a specific configuration. Since it represents probabilities, we have $\sum_{i=0}^{2^{2m}-1} \pi_i = 1$. A configuration $\mathcal{C}_i$ is given as the binary representation of its index i. Given, for example, a quantum repeater with one memory per party, four configurations $\mathcal{C}_0 = \{00\}, \mathcal{C}_1 = \{01\}, \mathcal{C}_2 = \{10\}, \mathcal{C}_3 = \{11\}$ exist. If all four possible configurations can be found with equal probability, the corresponding state vector is $\pi = (0.25, 0.25, 0.25, 0.25)^T$. If all memories are empty, the vector looks like $\pi = (1,0,0,0)^T$.

A transition that gives the change between two different memory configurations (states) occurs with a certain probability. This depends on the initial configuration and the underlying physical setup. To fully describe the quantum repeater, we need two transitions: one describing the transition due to sending qubits to the repeater and one for the Bell state measurement performed at the quantum repeater. Each transition is given by a stochastic matrix called the *transition matrix* T. All entries T_{ij} describe the probabilities for a specific transition between two memory configurations, namely from configuration $\mathcal{C}_j$ to configuration $\mathcal{C}_i$. Therefore, each entry is a nonnegative real number (i.e., $T_{ij} \in [0, 1]$), and all entries of a column sum to one. Note that due to the binary representation of i, j for the configurations, we get $i, j \in \{0, 1, \ldots, 2^{2m} - 1\}$.

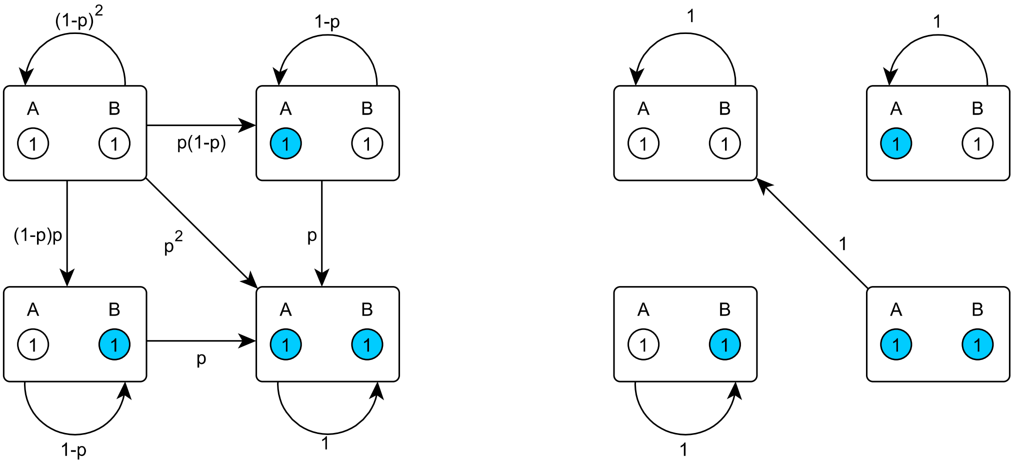
The first transition is described via the *storage map* $\sigma_\ell : \mathcal{H}_w^m(0) \to \mathcal{H}_w^m(\ell)$ with the set $\mathcal{H}_w^m(\ell)$ of all configurations that lead to ℓ Bell state measurements for a given m and w. This means the storage map takes the set of configurations for which no measurements can be performed (i.e., $\ell = 0$) to the set of configurations for which ℓ Bell state measurements can be performed. The second transition is given by the *measurement map* $\mu_l : \mathcal{H}_w^m(\ell) \to \mathcal{H}_w^m(0)$. Here, the transition goes from the set of configurations that lead to ℓ Bell state measurements to the set of configurations for which no measurements can be performed. Note that the measurement map directly depends on the connection length w. Taking for example the configuration from Fig. 2.8, it holds that $\{1010000110\} \in \mathcal{H}_2^5(2)$ but $\{1010000110\} \notin \mathcal{H}_1^5(2)$ since for a maximal connection length of $w = 1$, only one Bell state measurement can be performed.

We call the configuration at the beginning of a round $\mathcal{C}$. The storage map σ_ℓ gives the transition due to the successfully arriving qubits to the intermediate configuration $\mathcal{C}'$. Due to the Bell state measurements, the configuration changes to the final configuration denoted as $\mathcal{C}''$. This is given by the measurement map μ_ℓ. The following round starts with the new initial configuration $\mathcal{C}''$ of the previous round. The chronological process of one protocol round is as follows:

$$\begin{tikzcd} \mathcal{C}'(s-1) \arrow[r, "\mu_\ell"] & \mathcal{C}''(s-1)/\mathcal{C}(s) \arrow[r, "\sigma_\ell"] & \mathcal{C}'(s) \arrow[r, "\mu_\ell"] & \mathcal{C}''(s)/\mathcal{C}(s+1) \arrow[r, "\sigma_\ell"] & \mathcal{C}'(s+1) \end{tikzcd}$$

$$\begin{tikzcd} {} \arrow[r, dashed, "\text{Protocol round } s-1"'] & {} & {} \arrow[r, "\text{Protocol round } s"'] & {} & {} \arrow[r, dashed, "\text{Protocol round } s+1"'] & {} \end{tikzcd}$$

The storage map σ_ℓ gives the transition according to the success probability p



((a)) Storage map σ_ℓ ((b)) Measurement map μ_ℓ

Figure 2.9: ([KTW+25]) Transition maps for the quantum repeater with one memory per party. Along the arrows, the probabilities for the transitions are given. Arrows between two memory configurations that are not shown have zero probability. Memories shown in blue indicate that they are filled with a qubit.

of the qubits when sent through the channel (see Eq. (2.53)). The probability that a qubit arrives at the quantum repeater station and the configuration therefore changes from $\mathcal{C}$ to $\mathcal{C}'$ is given by

$$\begin{aligned} Prob[\sigma_\ell(\mathcal{C}) = \mathcal{C}'] &= Prob[\mathcal{C}'|\mathcal{C}] \\ &= \prod_i Prob[c_i'|c_i], \end{aligned} \tag{2.55}$$

with

$$Prob[c_i'|c_i] = (1-p)(1-c_i')(1-c_i) + pc_i(1-c_i') + c_ic_i' \tag{2.56}$$

for each bit $c_i, c_i' \in \{0,1\}$ from the bit string $\mathcal{C}$ and $\mathcal{C}'$, respectively, each representing the configuration of a single memory. Fig. 2.9(a) shows the graphical representation of the storage map in a quantum repeater with one memory per party. In that example, the quantum repeater can take one of the four configurations: $\mathcal{C}_0 = \{00\}, \mathcal{C}_1 = \{01\}, \mathcal{C}_2 = \{10\}$, and $\mathcal{C}_3 = \{11\}$. From Eq. (2.55), the transition probabilities between the four memory configurations follow. The matrix that represents the storage map is given by:

$$\sigma_\ell = \begin{pmatrix} (1-p)^2 & 0 & 0 & 0 \\ (1-p)p & 1-p & 0 & 0 \\ p(1-p) & 0 & 1-p & 0 \\ p^2 & p & p & 1 \end{pmatrix}. \tag{2.57}$$

The column of the matrix gives the starting configuration $\mathcal{C}$, and the row the final configuration $\mathcal{C}'$. Entry $(\sigma_\ell)_{20} = p(1-p)$, for example, gives the probability for the transition from the initial configuration $\mathcal{C}_0$ to the intermediate configuration $\mathcal{C}'{}_2$, i.e., $Prob[10|00]$. In detail, we find the probability that a memory remains empty as $Prob[c' = 0|c = 0] = 1 - p$, and the probability that a memory gets filled as $Prob[c' = 1|c = 0] = p$. The total probability of transitioning from configuration $\mathcal{C}_0$ to configuration $\mathcal{C}_2'$ is then given by the product, i.e., $Prob[\sigma_\ell(\mathcal{C}_0) = \mathcal{C}_2'] = p(1-p)$.

The transition due to the Bell state measurement is generally connection length-dependent. Which transition follows from which configuration $\mathcal{C}' \in \mathcal{H}_w^m(\ell)$ is determined by the matching problem described previously. Independent of the choice of the matching, it holds that one configuration either changes to another configuration because the measurement is performed or no change is made. The mapping is previously defined and unique. Additionally, we assume the Bell state measurement to be perfect, i.e., $p_{BSM} = 1$. Therefore, we find $(\mu_\ell)_{ij} = 1$, if a measurement is performed, and $(\mu_\ell)_{ij} = 0$ otherwise. In the example shown in Fig. 2.9(b), a Bell state measurement can be performed only on configuration $\mathcal{C}_3 = \{11\}$. There is no matching for all other configurations, and the configuration remains unchanged. The resulting matrix that defines the measurement map is therefore given by

$$\mu_\ell = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.58}$$

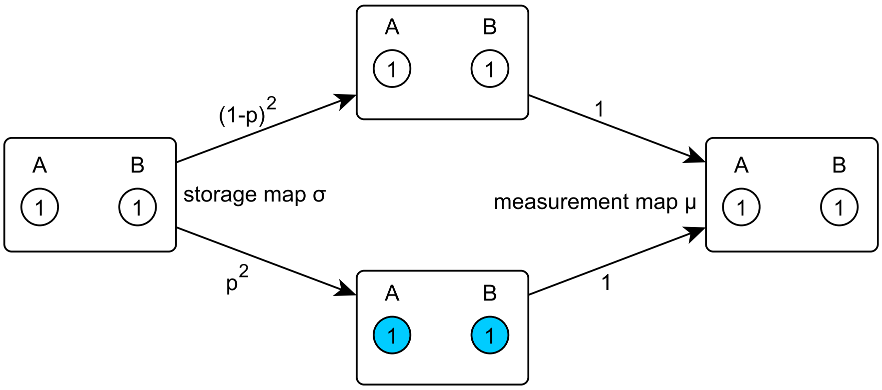


Figure 2.10: Possible paths to get from the memory configuration $\mathcal{C}_0$ back to the same configuration within one round.

To get the transition of one protocol round, i.e., combining the storage and the measurement map, one has to concatenate both transition matrices

$$T = \mu_\ell \circ \sigma_\ell, \tag{2.59}$$

which is again a valid transition matrix. In our example, that leads to the following transition matrix:

$$T = \mu_\ell \circ \sigma_\ell = \begin{pmatrix} (1-p)^2 + p^2 & p & p & 1 \\ (1-p)p & 1-p & 0 & 0 \\ p(1-p) & 0 & 1-p & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \tag{2.60}$$

All entries of the total transition matrix T remain nonnegative real numbers, and all columns sum to one. Each entry of the matrix gives the total probability to go from any configuration $\mathcal{C}_j$ at the beginning of a round to another configuration $\mathcal{C}''_i$ at the end of this round. Note that there can be more than one transition from configuration $\mathcal{C}_j$ to configuration $\mathcal{C}''_i$. Fig. 2.10 shows exemplary all different paths illustrating the transitions that can occur to get from configuration $\mathcal{C}_0$ back to $\mathcal{C}''_0$ in a single round. Since probabilities along a path are multiplied, and probabilities of all parallel paths are summed, we get the same total probability in Fig. 2.10 and in the corresponding entry T_{00} of the transition matrix.

Now, we can calculate the state vector π for the end of each round to get the probabilities π_i of every configuration $\mathcal{C}_i$. For the initial configuration, we start with empty quantum memories, i.e., $\pi_{init} = \begin{pmatrix} 1 & 0 & \dots & 0 \end{pmatrix}^T$, as no qubits are sent to the quantum router before starting the protocol. The probability distribution π_s after one round s is then given by

$$\begin{aligned} \pi_s &= T\pi_{init} \\ &= (\mu \circ \sigma)\pi_{init}. \end{aligned} \tag{2.61}$$

Considering a specific memory configuration $\mathcal{C}$, the transition to another memory configuration $\mathcal{C}''$ is calculated as follows:

$$Prob[\mu_\ell \circ \sigma_\ell(\mathcal{C}) = \mathcal{C}''] = \sum_{\mathcal{C}' \in \mathcal{H}_w^m(\ell)} \delta_{\mu_\ell(\mathcal{C}'),\mathcal{C}''} Prob[\sigma_\ell(\mathcal{C}) = \mathcal{C}'], \tag{2.62}$$

with Kronecker delta $\delta_{x,x'} = 1$ for $x = x'$ and $\delta_{x,x'} = 0$ otherwise. Eq. 2.62 is used to calculate a specific entry of the state vector π, i.e., the entry π_i for a given configuration $\mathcal{C}_i$. The probability to find any configuration $\mathcal{C}$ at the beginning of a round s (that equals the configuration $\mathcal{C}''$ at the end of the previous round $s-1$) depends on the initial configuration $\mathcal{C}$ of the previous round $(s-1)$:

$$Prob[\mathcal{C}(s)] = \sum_{\mathcal{C}(s-1) \in \mathcal{H}_w^m(0)} \sum_{\ell=0}^{m} Prob[\mu_\ell \circ \sigma_\ell\left(\mathcal{C}(s-1)\right) = \mathcal{C}''(s-1)] Prob[\mathcal{C}(s-1)]. \tag{2.63}$$

To calculate the probability of performing ℓ Bell state measurements on the intermediate configuration $\mathcal{C}'(s)$ of a round s ($Prob[\Lambda = \ell](s)$), we need to apply the storage map once more to the initial configuration $\mathcal{C}(s)$:

$$\begin{aligned} Prob[\Lambda = \ell](s) &= \sum_{\mathcal{C}' \in \mathcal{H}_w^m(\ell)} Prob[\mathcal{C}'(s)] \\ &= \sum_{\mathcal{C}' \in \mathcal{H}_w^m(\ell)} \sum_{\mathcal{C} \in \mathcal{H}_w^m(0)} Prob[\sigma_\ell(\mathcal{C}) = \mathcal{C}'] Prob[\mathcal{C}(s)], \end{aligned} \tag{2.64}$$

with the random variable Λ that can take values from 0 to m. Analogously, we proceed in the Markov chain picture:

$$\begin{aligned} Prob[\Lambda = \ell] &= \sum_{\mathcal{C}'_i \in \mathcal{H}^m_w(\ell)} \pi_i \\ &= \sum_{\mathcal{C}'_i \in \mathcal{H}^m_w(\ell)} \left(\sigma_\ell \pi_{init}\right)_i , \end{aligned} \tag{2.65}$$

with state vector π_{init} giving the probability distribution at the beginning of the round. With this probability, it is now possible to calculate the probability of having ℓ successful Bell measurements:

$$Prob[\Sigma = \ell] = \sum_{i=\ell}^{m} \binom{i}{\ell} Prob[\Lambda = i](s) p^{\ell}_{BSM} \left(1 - p_{BSM}\right)^{i-\ell} , \tag{2.66}$$

with random variable $\Lambda \in [0, m]$ as the number of performed Bell state measurements and random variable $\Sigma \in [0, m]$ as the number of successfully performed Bell state measurements. Whether a Bell state measurement is successful is given by the success probability p_{BSM}. Finally, we calculate the average number of successfully performed Bell state measurements as

$$\langle \ell \rangle = \sum_{\ell=0}^{m} \ell Prob[\Sigma = \ell](s). \tag{2.67}$$

The repeater rate can now be calculated by combining all the equations and finally inserting Eq. (2.67) into Eq. (2.54).

2.3.2 Multipartite quantum router

So far, we have considered only the bipartite setup, where the quantum repeater is placed between two end nodes. Now, we consider the multipartite setup. In our work, we focus on the star-shaped network with one central station and all end nodes located around it. Due to its structure, we call the central station the quantum router. The task of the quantum router is the same as that of the quantum repeater: the quantum router is placed in the middle between the end nodes to overcome larger distances. It uses entanglement swapping to distribute entanglement between all end nodes. As described in Sec. 2.2.2, the entanglement swapping is performed using a GHZ state measurement. Consequently, the qubits held locally are projected onto one of the GHZ states given in Eq. 2.51. With classical communication and local operations, the desired GHZ state can be obtained.

In [CKD$^+$21, VGNT21b], similar graph structures are analyzed analytically and numerically with the goal of entangling $k \leq N$ parties. Li et al. consider multipartite star graphs without quantum memories [LFL$^+$23b]. Here, we introduce quantum memories into the quantum router and generalize the memory multiplexing to the multipartite setup. In this section, we give a brief overview of the setup of a quantum router and introduce memory multiplexing. A detailed discussion of the underlying multipartite matching, as well as the multiplexing strategies, is given in Chapter 3 and Chapter 4 when we present our results.

Multiplexing

With memory multiplexing, the GHZ state generation rate can be increased similarly to that of the bipartite scenario. Fig. 2.11 shows an example setup of a quantum router with four parties and five memories each. The parties are here called Alice, Bob, Charly, and Dave. Due to the underlying quantum circuit of performing a GHZ state measurement, the connections within the quantum router are made between $N-1$ parties (called peers B_i with $i \in \{1, 2, \ldots, N-1\}$) and one party, called central party A (compare Sec. 2.2.2 for details). Here, Alice (A) is considered to be the center node, while Bob (B_1), Charly (B_2), and Dave (B_3) are the peers.

Again, all memories are labeled so that the connection length can again be defined as the difference in the memory labels. Note that for the multipartite setup, this is an artificial construction since the labels are made randomly and depend on the arrangement of the memories made in the experimental setup. However, the connection length is considered to include the assumption that it is impossible to always connect all memories from one party with all memories from another party.

Similar to the bipartite setup, the choice of the memories used for the GHZ state measurement is based on a matching problem in graph theory. We properly define

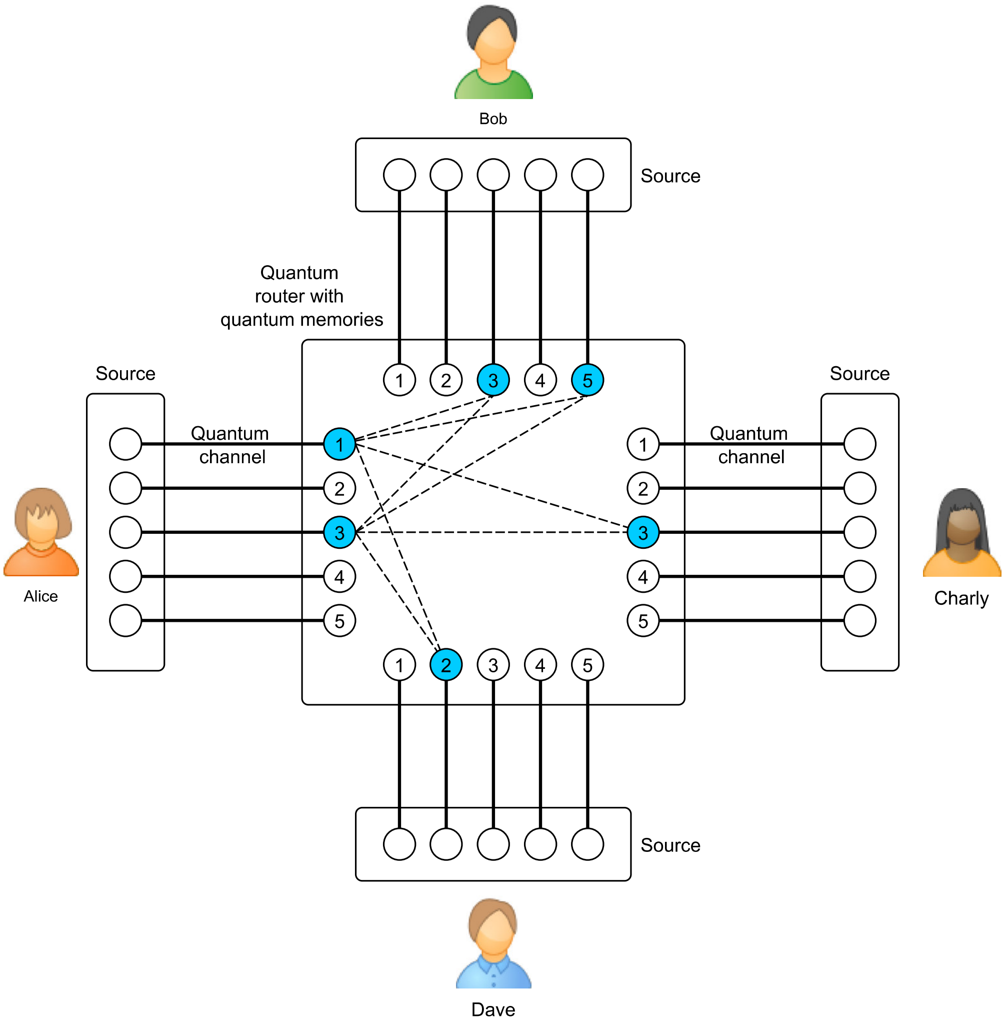


Figure 2.11: ([KKB24]) Representation of a four-partite quantum router with multiplexing. Up to five GHZ state measurements can be performed in parallel, as every party holds five photon sources and has five quantum memories available. Quantum memories that are shown in light blue represent filled memories. Possible connections between the memories are indicated by the dashed lines.

the *multipartite matching* that shows up in quantum routers and give a concrete description of the graph theoretical problem and its complexity in Chapter 3.

Router rate

Analogously to the repeater rate, we define the router rate of a quantum router as the number of GHZ state measurements performed successfully on average per round and memory:

$$R(s_c) = \frac{1}{s_c} \sum_{s=1}^{s_c} \frac{\langle \ell \rangle}{m}. \tag{2.68}$$

The calculation of the router rate is similar to the one for the quantum repeater given in Sec. 2.3.1. However, the bit strings $\mathcal{C}$ describing the memory configurations of the quantum router with N parties and m memories per party are now of length Nm. In total, there are now 2^{Nm} different memory configurations the quantum router can be in. Therefore, the probability distribution given by the state vector π as well as the two transition matrices σ_ℓ and μ_ℓ are now of dimension 2^{Nm} and $2^{Nm} \times 2^{Nm}$, respectively. The entries of the storage map σ_ℓ are again calculated via Eq. (2.56) while the measurement map μ_ℓ relies on the underlying multipartite matching.

2.4 Quantum key distribution

We now focus on an application in quantum communication: the *quantum key distribution.* For this purpose, we consider the following situation:

Alice and Bob - one of whom is located in Duesseldorf and the other one in Munich - want to share their secret information about their bank accounts. This is a message in binary form: "0110100101010001101110". For security reasons, they do not want to convey this information on the phone since any eavesdropper (a malicious third party that is not authorized to get this information) could listen to them. The same holds for any other communication method. Any letter, either written on paper or by electronic mail, could be intercepted, copied, or just read in between. To increase security, Alice and Bob agree to use a secret key that only they know. This key

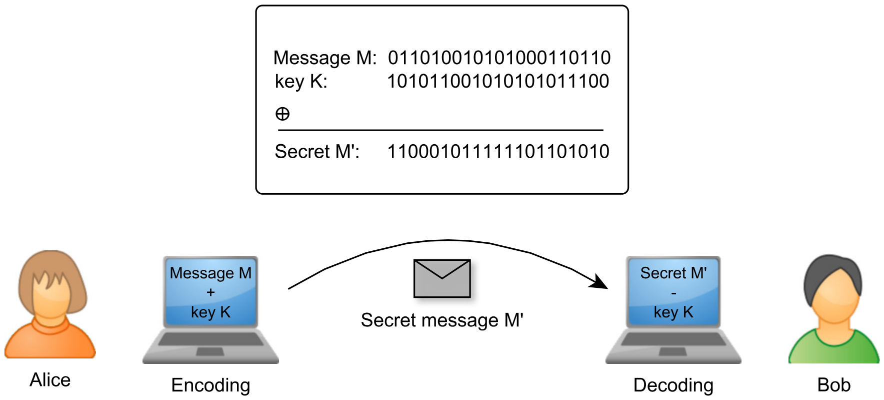

Message M: 0110100101010001101110
key K: 1010110010101010111100
⊕
Secret M': 1100010111111101101010
Message M
+
key K
Secret message M'
Secret M'
-
key K
Alice
Encoding
Decoding
Bob

Figure 2.12: Illustration of secure communication using a secret key. Alice encodes the message by adding a secret key (addition modulo 2: ⊕), and Bob decodes the message by subtracting the same key.

consists of a string of uniformly random bits at least as long as the message. In this example, the key is given by ”1010110010101010111100”. The party who wants to send the message, let us say Alice, can use the secret key to encode her message: she adds the secret bit string bitwise (addition modulo 2) to the original message to create another bit string from which the information cannot be generated without further knowledge about the secret key. Conversely, Bob can subtract the same secret bit string to recover the original message. This is the decoding. An example scheme is shown in Fig. 2.12.

The encoded message can be sent over a public channel. Even if it is intercepted by a third party, the information is secure since no reference to the original message is given. Only the parties that have knowledge about the secret key can recover the information. This encryption method, using a pre-shared key only once, is called *one-time-pad* [Sha49]. It is proven to be secure under the following assumptions:

1. The key is generated truly randomly.
2. The key and all subparts are used only once.
3. All parties keep the key secret and do not share it with anyone.
4. The key is at least as long as the message that is to be encrypted.

Now that Alice and Bob know how they can send the message secretly, they have a new challenge to deal with: *how can they share the key secretly so that no eavesdropper gets to know the key?* In classical computation, *public key cryptography* [DH76] such as *RSA* [RSA78] is a common technique used nowadays. That is asymmetric cryptography, in which the keys for encoding and decoding are different. One party publishes a key, let us say Alice. Every party that wants to send a message to Alice can use the public key for encryption. The key for decryption is a different secret key only known to Alice. She uses that key to decrypt the message and recover the original information. In that way, the distribution of a symmetric secret key that all communicating parties need for encryption and decryption can be performed.

The RSA cryptosystem, like many other public key cryptosystems, uses large prime numbers. The security proof of RSA then relies on the complexity of factorizing large numbers: given a number n, the task is to find the product of prime numbers that equals n. Although the test of deciding whether a number is prime or not is in $\mathcal{P}$ [AKS02], it is not known whether the factorization problem is in $\mathcal{P}$. So far, no efficient algorithms exist to solve this task if the key has enough bits [2]. An overview of classical key distribution and the methods of symmetric and asymmetric keys is given, for example, in [Rot05].

With the development of quantum computers, the long-term security of public key cryptography is no longer guaranteed. Shor's algorithm is said to be efficient in solving the factorization problem [Sho94, Sho97]. However, quantum technologies also lead to new opportunities: one can make use of the properties of quantum mechanics to solve the task of distributing a secret key. Quantum key distribution offers an information-theoretically secure solution to share a secret key between the parties. We first consider the *BB84 protocol*, named after its inventors Bennett and Brassard in 1984 [BB84].

[2] Since January 2025, for example, the German Federal Office for Information Security has recommended a modulus $n = pq$ (with primes p and q) of at least 3000 bits in length [Fed24].

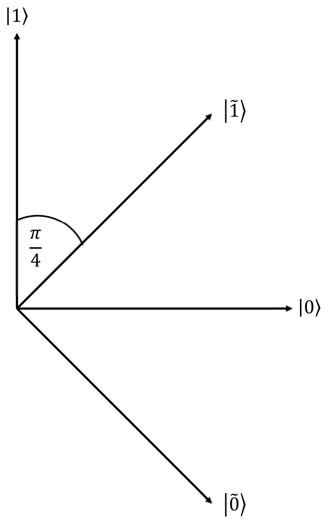


Figure 2.13: Representation of the two orthogonal bases used to encode the bits for the key distribution: $0 \mapsto |0\rangle$ or $|\tilde{0}\rangle$, $1 \mapsto |1\rangle$ or $|\tilde{1}\rangle$.

2.4.1 BB84 protocol

We return to Alice and Bob, who still have to handle the problem of secretly sharing a key. So far, they have shifted the problem of sharing a message secretly to the problem of distributing the key without being eavesdropped on. To tackle that problem, they want to perform the bipartite BB84 protocol to distribute a random secret key that they can use to share their private message about the bank account. To do so, they make use of two orthogonal bases to encode a classical bit:

$$\mathcal{B}_1 = \{|0\rangle, |1\rangle\} \tag{2.69}$$

$$\mathcal{B}_2 = \{|\tilde{0}\rangle, |\tilde{1}\rangle\}. \tag{2.70}$$

Photons are used to set up the qubits physically. The information is encoded in the polarization, and the basis defines the orientation of the polarization (compare Fig. 2.13). The steps of the protocol are as follows:

1. **Prepare-and-measure:** Repeat the following steps n times:

   a) Alice randomly chooses a basis $\mathcal{B}_1$ or $\mathcal{B}_2$ and polarizes the photon either horizontally ($|0\rangle, |\tilde{0}\rangle$) or vertically ($|1\rangle, |\tilde{1}\rangle$).

b) The qubit is sent through a quantum channel. When Bob receives the photon, he randomly chooses a basis and measures the polarization.

2. **Sifting:** Alice tells Bob which basis she chose in every round over the authenticated classical channel. Rounds in which the basis choice of Alice and Bob did not coincide are discarded.

Alice and Bob did perform the protocol up to that step. The following table shows an example of what they came up with:

Alice's basis	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$
Prepared photon	$\rightarrow$	$\rightarrow$	$\nearrow$	$\uparrow$	$\searrow$	$\nearrow$	$\searrow$	$\uparrow$	$\rightarrow$	$\searrow$
Bob's basis	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$
Measurement outcome	r	$\rightarrow$	$\nearrow$	$\uparrow$	r	$\nearrow$	$\searrow$	$\uparrow$	r	r
Raw key		0	1	1		1	0	1		

In the rounds in which the basis choices differ, the measurement outcome is random, here indicated by the "r". Statistically, Alice and Bob choose the same basis for half of the rounds. Therefore, the raw key is of length approximately $n/2$.

The presence of an eavesdropper could change the setup as follows, depending on the power she has. One strategy the eavesdropper can choose is to intercept the qubit in the channel, perform a measurement on it, and resend the qubit to Bob. Since the announcement about the basis choice is made via a classical channel, the eavesdropper can access that information. Analogously to Bob, she has to discard all rounds in which she chose a basis other than the one Alice used for preparation. Nevertheless, her interaction with the qubit affects the state of the quantum system. The measurement outcome is random in all rounds in which the eavesdropper chooses a basis different from Alice's. Consequently, the state of the qubits is randomly set to one of the polarizations. This introduces an error, which can be later identified:

Round	1	2	3	4	5	6	7	8	9	10
Alice's basis	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$
Prepared photon	$\rightarrow$	$\rightarrow$	$\nearrow$	$\uparrow$	$\searrow$	$\nearrow$	$\searrow$	$\uparrow$	$\rightarrow$	$\searrow$
Eve's basis	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$
Resent photon	r	r	$\nearrow$	$\uparrow$	$\searrow$	r	r	r	$\rightarrow$	$\searrow$
Bob's basis	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_2$	$\mathcal{B}_1$	$\mathcal{B}_2$	$\mathcal{B}_1$
Measurement outcome	r	r*	$\nearrow$	$\uparrow$	r	r*	r*	r*	r	r
Raw key		r*	1	1		r*	r*	r*		

In all rounds in which Alice and Bob choose the same basis, but the eavesdropper chooses the opposite basis, Bob's measurement may differ from what Alice would expect. Due to Eve's measurement, the superposition of the photon collapses. With a probability of 1/2, she resends a photon with a polarization opposite to the one Alice originally sent. This random output ”r*” introduces an error. With a probability of 1/2, that bit in the raw key differs for Alice and Bob. This is to be verified in the next step of the protocol:

3. **Parameter estimation:** Alice and Bob use part of the raw key to estimate the *quantum bit error rates* (QBER) Q_X and Q_Z. Q_X gives the error from an X-basis measurement, while Q_Z gives the error from the measurement performed in the Z-basis. In that step, the presence of an eavesdropper can be inferred. Considering Eve's attack shown above, about $n/8$ of the key is corrupted on average: with a probability of 1/2, Alice and Eve choose different bases. With another chance of 1/2, the measurement result of Eve collapses the state to the polarization Alice did not choose. Due to Bob's basis choice, the round is discarded with a probability of 1/2. Whenever Alice and Bob compare parts of the key and find such a large difference in the key, the protocol should be aborted and has to be started again.

Additional steps of the BB84 protocol are:

4. **Information reconciliation:** Alice and Bob perform some classical error correction on the raw key to compensate for potential errors arising from the interaction of the qubit with its environment. Communication takes place via the public authenticated channel. At the end of this step, both parties hold the same bit string.

5. **Privacy amplification:** In this step, Alice and Bob agree on a hash function they both apply to their bit strings to generate a shorter but completely secret bit string: the final key K.

The efficiency of the protocol can be improved by choosing one basis with a higher probability. The rounds in which this preferred basis is used for key generation, while the rounds using the other basis are used for testing. This technique does not compromise security [LCA05].

Entanglement-based version

Instead of using single-photon sources that can prepare single photons and send them one by one, we can use an entanglement-based version of the BB84 protocol [BBM92]. In this protocol, the prepare-and-measure part is changed as follows:

1. **Prepare-and-measure:** Repeat the following steps n times:
   a) An entangled state (ideally the maximally entangled Bell state $|\phi^+\rangle$) is distributed between the parties Alice and Bob by any source.
   b) Alice and Bob each choose a measurement basis in which they measure their qubit of the Bell pair. Both parties record their measurement outcome.
2. All other steps remain the same.

Note that the distribution of an entangled state can be done via a quantum repeater performing entanglement swapping. Both parties then hold one qubit of the entangled pair on which they both can perform a measurement, recording their measurement outcome.

Security of the BB84 protocol was proven in [SP00] with a simple proof based on entanglement purification. A more general proof is given, for example, in [KGR05]. In general, the eavesdropper can perform different attacks, depending on the power she has. For the different attacks, different security proofs are needed. Additionally, some assumptions have to be made in order to guarantee security in practical QKD. Since we do not deal with the security of quantum key distribution in this work, we leave it at this point and refer the reader to [SBPC$^+$09] for further details.

2.4.2 Multipartite BB84 protocol

A secret key can also be distributed among a larger set of parties. We call this scenario *conference key agreement*. Sequentially performing the bipartite quantum key distribution protocol can solve this task. However, due to multipartite quantum correlations, more efficient protocols can be designed [MGKB20]. Here, we focus on the multipartite BB84 protocol (also called N-BB84 protocol). The protocol makes use of the multipartite entangled GHZ state (for N parties)

$$\left|GHZ_0^+\right\rangle = \frac{1}{\sqrt{2}} \left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right), \tag{2.71}$$

because the measurement outcomes in the Z-basis provide perfect correlation and are random and uniformly distributed.

The steps of the protocol are a generalization of the protocol steps from the entanglement-based bipartite BB84 protocol given above:

1. **Prepare-and-measure:** Repeat the following steps n times:

   a) A quantum source distributes a multipartite entangled GHZ state among all parties so that each party holds one qubit of the state.

   b) Each party performs a measurement on their own qubit and records the outcome. The measurement is performed in one of the two bases (i.e., key generation or test basis).

All other steps remain the same. The parties can perform the **sifting** step in order to discard rounds in which the measurement bases chosen by the parties differ. The QBERs are estimated in the **parameter estimation** step. Afterward, some error correction protocol is applied in the **information reconciliation**, and lastly, the key is shortened to a completely secure key in the **privacy amplification**.

Since key generation only works if all N parties choose the same measurement basis in one round, this basis is again chosen with a higher probability than the test basis.

2.4.3 QBER and key rate

The information we gain during the conference key agreement is given by the quantum bit error rates estimated in the parameter estimation step. The QBER in the X-basis gives the error with respect to the multipartite entangled GHZ state from Eq. 2.71. If the state held by the parties equals the GHZ state, the QBER is zero. Otherwise, it gives the probability that the state differs from the GHZ state. Explicitly, Q_X is calculated as

$$Q_X = \frac{1 - \langle X^{\otimes N} \rangle}{2}. \tag{2.72}$$

The expectation value of the X-operator is given by $\langle X \rangle = tr\,(\rho X)$ for a density matrix ρ. The QBER in Z-basis gives the probability that one of the parties B_i gets a different measurement outcome than party A does. In the multipartite setup, we call this the bipartite QBER Q_{AB_i}. Explicitly, it holds:

$$Q_{AB_i} = \frac{1 - \langle Z_A Z_{B_i} \rangle}{2}. \tag{2.73}$$

With the QBER, the *asymptotic secret fraction* can be calculated. This rate gives the fraction of secret bits that are preservable from all measured bits. For the N-BB84 protocol, [GKB18] gives the asymptotic secret fraction as follows:

$$r_\infty = 1 - h(Q_X) - \max_{1 \leq i \leq N-1} h(Q_{AB_i}). \tag{2.74}$$

For the bipartite setup, we find

$$r_\infty = 1 - h(Q_X) - h(Q_Z), \tag{2.75}$$

with Q_Z being the probability that party A and B get different measurement outcomes for the Z-basis measurement. In both equations, $h(p) = -p\log_2(p) - (1 - p)\log_2(1-p)$ is the *binary Shannon entropy* function.

The secret key rate is defined as the fraction of all distributed bits that successfully contribute to the final key as secret bits. It is given by the product of the asymptotic secret fraction from Eq. 2.74 and the router rate from Eq. 2.68:

$$K(s_c) = r_\infty R(s_c). \tag{2.76}$$

CHAPTER 3

THE QUANTUM ROUTER MATCHING PROBLEM

3.1 Introduction

To implement the quantum router with memory multiplexing, it is essential to understand the underlying graph-theoretical problem of matching. It is of great interest to quickly decide how the quantum memories of the different parties have to be connected via a GHZ state measurement to optimize the distribution rate of entangled states and increase the secret key rate by reducing the storage time of the qubits in the memories. Efficient algorithms that solve the quantum router instances and the multiplexing requirements are to be developed here. The effect of the quantum memories on the fidelity of the stored qubits, as well as the underlying strategies that optimize the secret key rate, are discussed and evaluated in Chapter 4.

In our work, we focus on the quantum router matching. We give a proper definition and compare the problem with the known *N-dimensional matching* that is given in the literature. We further analyze the complexity of the quantum router matching. We start with the decision problem and arbitrary connections that can be chosen within the requirements given by the underlying quantum circuit. It turns out that this most general decision problem is $\mathcal{NP}$-complete, just as the known N-

dimensional matching is [Kar72]. From that, it follows that finding a matching with maximum cardinality (the maximum quantum router matching) cannot be easier. However, drawing the edges with respect to the connection length, which is physically motivated, leads to some special cases that are efficiently solvable. In our work, we examine different cases and give algorithms for those that are efficiently solvable. Furthermore, we propose an approximate solution for the maximum quantum router matching in the general case using a greedy algorithm. For small input graphs, it is also possible to get exact solutions for the quantum router matching since the difference between exponential and polynomial runtime is not that significant. This can be used to compare the approximation algorithm for small graphs with an exact solution. For large inputs, however, the exponential runtime makes it impossible to find a maximum cardinality matching by checking all possible solutions. In that case, the approximation algorithm is needed.

3.2 Computational complexity

As this topic requires some background knowledge about computer science, we first introduce the concept of *computational complexity.* This measures how many resources are needed to solve a problem algorithmically. Considering a *Turing machine* (TM), there are two resources considered: the *time* (number of computation steps) and the *space* (number of cells on the working tape). A Turing machine is a theoretical computation model that forms the basis of a modern computer. It is named after its inventor, Alan Turing [Tur37, Tur96]. Equipped with an infinite tape, a read/write head, and a set of rules that determine its actions based on the current state and tape symbol, it can simulate any algorithm. A *nondeterministic TM*, in contrast to a *deterministic TM*, can choose between multiple options on how to proceed. That is, it can have several paths in its computation. Acceptance is reached if and only if at least one accepting path exists. Deterministic TMs, however, cannot have nondeterministic paths. But they can just go through all paths of the nondeterministic TM sequentially, such that they are just as powerful as nondeterministic TMs. Note that this takes up to exponentially more computation steps.

In the following, we focus on the resource time. Regarding deterministic Turing machines, the computation time is the number of computation steps that need to be performed until a computation is done. For a nondeterministic TM, the number of steps required to perform a computation is the shortest accepting path. If such a path does not exist, the computation time is undefined. In general, the computation time of an instance is related to the input size. To classify problems, *complexity classes* are used. Each class contains problems that can be solved by a given maximal resource (always with respect to the most efficient algorithm that solves the problem). Here, we focus on the time complexity classes $\mathcal{P}$ and $\mathcal{NP}$. A problem is considered to be *efficiently* solvable if a deterministic TM exists that can solve the problem in polynomial time. Any *decision problem* (leading to a 'yes' or 'no' output, i.e., either acceptance or no acceptance) that can be decided on a deterministic TM in polynomial time, leading to a correct answer, is in $\mathcal{P}$. Problems outside of $\mathcal{P}$ are called *intractable*, meaning that any deterministic TM requires running exponentially many computation steps with respect to the input size. A decision problem is in $\mathcal{NP}$, if and only if there exists a nondeterministic TM that runs in polynomial time and decides the problem. Note that $\mathcal{P} \subset \mathcal{NP}$ [1].

A decision problem is called $\mathcal{NP}$-hard if every problem in $\mathcal{NP}$ can be reduced to it in polynomial time. Moreover, a decision problem is called $\mathcal{NP}$-complete if it is both in $\mathcal{NP}$ and $\mathcal{NP}$-hard. In general, $\mathcal{NP}$-complete problems are said to be the "hardest" problems in this complexity class. Note that $\mathcal{P}, \mathcal{NP}$, and $\mathcal{NP}$-completeness are only defined for decision problems, i.e., only for problems that have a yes/no answer. The Euler diagram representing these two complexity classes is given in Fig. 3.1. Besides these two complexity classes, there are more classifications regarding time and space resources. A more detailed overview of computational complexity is given, for example, in the book by Rothe [Rot05].

3.3 Quantum router matching

Similar to the bipartite quantum repeater with memory multiplexing, the quantum router setup is also based on a matching problem from graph theory. The quantum memories in the router form a graph as in the bipartite setup. However, since there

[1] The question of whether $\mathcal{P} = \mathcal{NP}$ remains open. However, it is generally assumed that they are not equal.

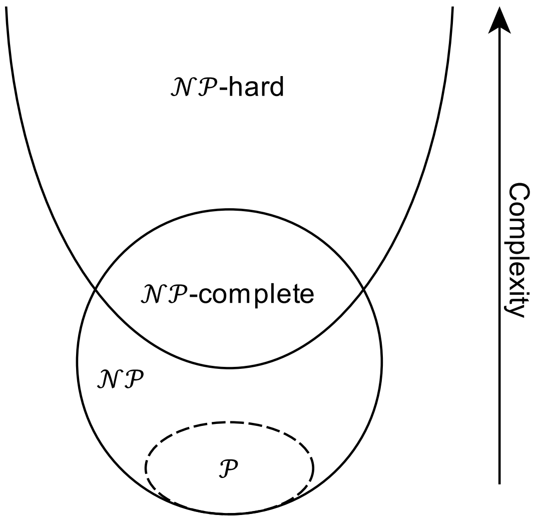


Figure 3.1: Euler diagram for the set of $\mathcal{P}$, $\mathcal{NP}$, $\mathcal{NP}$-hard, and $\mathcal{NP}$-complete problems under the assumption that $\mathcal{P} \neq \mathcal{NP}$.

are now N parties, i.e., N subsets of memories, this is an N-partite graph $G = (V, E)$. The set of nodes V is divided into N pairwise disjoint subsets $V_1, V_2, \dots, V_N$. Again, the edges indicate the possible connectivity between those nodes. Due to the graph structure, the problem we consider here is a special form of *N-dimensional matching.*

The N-dimensional matching known from literature is defined on *hypergraphs.* A hypergraph is a pair (V, E) with nodes V and hyperedges E that is given by a set of any number of vertices $v \in V$. An example hypergraph is given in Fig. 3.2. The N-dimensional matching is defined as follows [LP09]:

Definition 3.1 *(N-dimensional matching). Given the N finite sets V_ι with $\iota \in \{1, \dots, N\}$ and let T be a subset of $V_1 \times V_2 \times \cdots \times V_N$. That means that T consists of N-tuples $(v_1, v_2, \dots, v_N)$ with $v_1 \in V_1, v_2 \in V_2, \dots, v_N \in V_N$. The set $M \subseteq T$ is an N-dimensional matching if for any two distinct N-tuples $(v_1, v_2, \dots, v_N) \in M$ and $(\tilde{v}_1, \tilde{v}_2, \dots, \tilde{v}_N) \in M$ it holds $v_1 \neq \tilde{v}_1, v_2 \neq \tilde{v}_2, \dots, v_N \neq \tilde{v}_N$.*

The corresponding decision problem of this verifies whether a matching exists with at least cardinality γ for a given input hypergraph and a positive integer γ. Already, the 3-dimensional matching (from Def. 3.1) as a decision problem was mentioned to

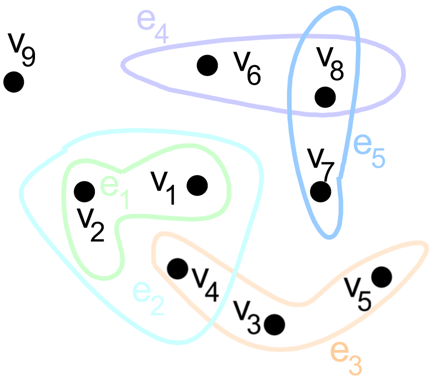


Figure 3.2: Example of a hypergraph with vertices $V = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8, v_9\}$ and hyperedges $E = \{e_1, e_2, e_3, e_4\}$ with $e_1 = \{v_1, v_2\}, e_2 = \{v_1, v_2, v_4\}, e_3 = \{v_3, v_4, v_5\}, e_4 = \{v_6, v_8\}, e_5 = \{v_7, v_8\}$.

be $\mathcal{NP}$-complete by Karp in 1972 [Kar72]. Thus, the optimization problem given by the maximum 3-dimensional matching that finds a maximum integer γ is not easily solvable, either. To prove $\mathcal{NP}$-completeness, a known $\mathcal{NP}$-complete problem is reduced to this problem. The full proof is given, for example, in [Har82, KT06]. There, the $\mathcal{NP}$-complete problem 3-SAT is reduced to the maximum 3-dimensional matching. Approximation algorithms can be used to solve this problem up to a certain convergence [Cyg13].

Let us now consider the matching we perform in the quantum router. Although the quantum router is defined on an N-partite graph, the connections between the individual parties are defined via bipartite edges and not as hyperedges. This is due to the underlying GHZ circuit performed at the quantum router (see Sec. 2.2.2 for details). Even though these edges can be grouped into hyperedges under certain criteria, the previous definition is not transferable. This is shown by a simple example in Fig. 3.3. The graph on the left of Fig. 3.3(a) shows a 3-partite graph with hyperedges including one node per party: $e_1 = \{v_B^1, v_A^2, v_C^2\}$ and $e_2 = \{v_B^3, v_A^2, v_C^1\}$

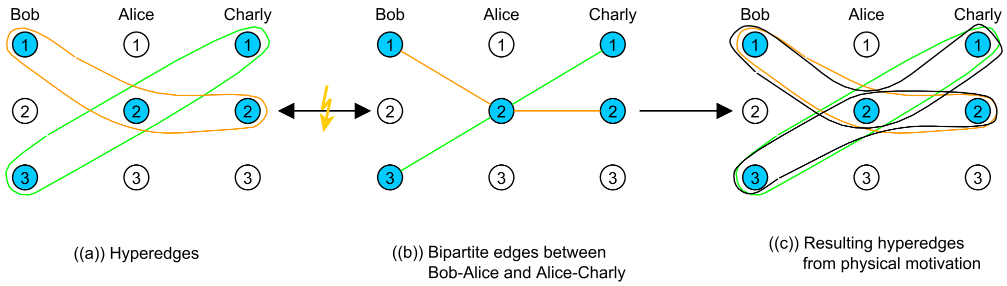

Bob
Alice
Charly
1
2
3
((a)) Hyperedges
((b)) Bipartite edges between Bob-Alice and Alice-Charly
((c)) Resulting hyperedges from physical motivation

Figure 3.3: Example of a 3-partite graph with (a) hyperedges and (b) bipartite edges forming a matching instance. The graph in (b) can be obtained by splitting the hyperedges into bipartite edges. The reconstructed hypergraph shown in (c) shows, however, that the quantum router is not described suitably by the hypergraph given on the left, as they are different from each other.

(with B: Bob, A: Alice, C: Charly). This forms a valid subset T of $V_A \times V_B \times V_C$ and is, therefore, a correct instance to which we can apply the 3-dimensional matching defined before. The graph in the middle of Fig. 3.3(b) shows the same graph instance but with bipartite edges between Bob and Alice, and between Alice and Charly. One can construct this graph from the graph on the left by splitting the hyperedges into bipartite edges. Nevertheless, we find two more hyperedges when constructing the hypergraph starting from the graph in the middle, in which the filled memories and allowed bipartite edges are given. The resulting hypergraph with the four hyperedges is shown on the right of Fig. 3.3(c). From a physical perspective, there is no reason why these two additional hyperedges (shown in black) should be excluded from the matching if the connection length is not restricted. This example shows that any graph instance we define in a quantum router is a special case of the general N-dimensional matching in graph theory. However, it is important to note that not all valid graph instances for the N-dimensional matching represent the physical setup in a suitable way.

We define the *quantum router matching* as follows:

Definition 3.2 *(Quantum router matching). Given an N-partite graph consisting of N disjoint vertex subsets $V_1, V_2, \ldots, V_N$ created from the given quantum router setup in each round. A quantum router matching M is a set of N-tuples ζ_k with*

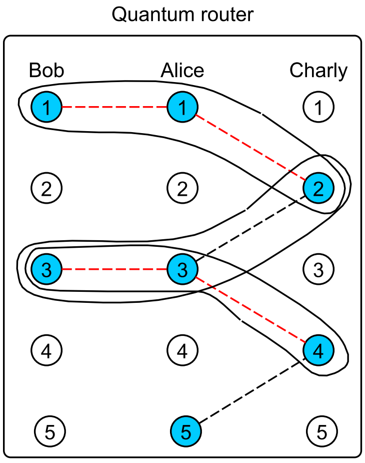


Figure 3.4: Example of a maximum quantum router matching in a 3-partite graph. The connection length is set to $w \leq 1$. The dashed lines show all possible bipartite edges between Bob and Alice and Alice and Charly. From that, all valid hyperedges are constructed. These are given by the solid lines. A possible maximum quantum router matching is shown in red.

$k \in \{1, 2, \ldots, |M|\}$ *, such that*

- *each N-tuple always consists of exactly one node from each subset V_ι, i.e., $(v_1, v_2, \ldots, v_N)$ with $v_1 \in V_1, v_2 \in V_2, \ldots, v_N \in V_N$,*
- *each N-tuple is created from bipartite edges between one special party, let us say center party V_N (or V_A) and all peers $V_1, \ldots, V_{N-1}$ (or V_{B_i} with $i \in \{1, \ldots, N-1\}$),*
- *no node is used more than once, and*
- *bipartite edges that do not conform to the connection length are not allowed.*

A quantum router matching is maximum when its cardinality $|M|$ is maximum, and no other choice of N-tuples can increase the cardinality.

An example of a tripartite graph with a valid maximum quantum router matching is shown in Fig. 3.4.

3.4 Results

In our work, we first define the corresponding decision problem and prove $\mathcal{NP}$-completeness for the most general case in which we draw edges randomly between filled quantum memories. Then we discuss special cases that result from integrating the connection length, for which we obtain efficient algorithms. In the case of the unweighted general quantum router maximum matching, we additionally provide an approximation algorithm.

3.4.1 Complexity results

For a complexity analysis, it is useful to formulate the problem as a decision problem first and check this for its hardness. If deciding whether some cardinality γ can be achieved is already hard, it shows that computing such a matching is hard, too. We define the general decision problem of the quantum router matching with arbitrary connections as follows:

Quantum router matching (QRM)

Given: An N-partite graph $G = (V, E)$ with N disjoint subsets with one central subset V_A and $i \in \{1, \dots, N-1\}$ peers V_{B_i}, and a positive integer γ.

Question: Does there exist a quantum router matching of at least size γ in G?

We further introduce a *Weighted multipartite maximum matching* that allows us to weight all vertices with respect to the fidelity of the qubits within the memories. A detailed discussion about different matching strategies is given in Chapter 4. In this work, we follow the strategy to always combine qubits with the highest fidelities, i.e., the lowest storage time (a physical motivation for that is also given in Chapter 4). For now, we assume that all weights W given by the fidelity of each node of one N-tuple (F_v with $v \in \zeta_k$) within a matching are multiplied, and the total weight of one matching $W(M)$ is given by the sum of the weights of all N-tuples $\zeta_K \in M$:

$$W(M) = \sum_{\zeta_k \in M} \prod_{v \in \zeta_k} F_v. \tag{3.1}$$

The resulting weighted multipartite matching is as follows:

Weighted quantum router maximum matching (WQRMM)

Given: An N-partite graph $G = (V, E)$ with N disjoint subsets with one central subset V_A and $i \in \{1, \dots, N-1\}$ peers V_{B_i}, the fidelity $F_v \in [0, 1]$ for each node $v \in V$, and a positive integer γ.

Question: Does there exist a maximum multipartite matching M with total weight at least γ in G?

We prove that the general decision problem given by QRM is – similar to the known N-dimensional matching – $\mathcal{NP}$-complete. To show $\mathcal{NP}$-hardness, we reduce to QRM from the well-known problem *3-SAT* to show that QRM is at least as hard as 3-SAT. This 3-SAT problem was independently proven $\mathcal{NP}$-complete by Cook [Coo71] and Levin [Lev73]. That QRM is in $\mathcal{NP}$ (and thus $\mathcal{NP}$-complete) follows immediately from the fact that any solution to the problem can be guessed nondeterministically in polynomial time, and its correctness can be shown deterministically. Since the unweighted maximum QRM for a general graph is intractable, it immediately follows that WQRMM is intractable as well.

Although intractability holds for the general cases of the introduced QRM (WQRMM) with arbitrary connections, we give special cases in which the problems are efficiently solvable:

1. QRM is in $\mathcal{P}$, whenever every node from the peers V_{B_i} with $i \in \{1, \dots, N-1\}$ is connected to at most one node from the center party V_A.

2. WQRMM is in $\mathcal{P}$, whenever every node from the peers V_{B_i} with $i \in \{1, \dots, N-1\}$ is connected to at most one node from the center party V_A.

3. QRM is in $\mathcal{P}$, whenever every node from the peers V_{B_i} with $i \in \{1, \dots, N-1\}$ is connected to every node within the center party V_A.

4. WQRMM is in $\mathcal{P}$, whenever every node from the peers V_{B_i} with $i \in \{1, \dots, N-1\}$ is connected to every node from the center party V_A.

5. QRM is efficiently solvable via *Network Flow* for all tripartite graph instances.

Note that the results 1 and 2 are both independent of the connection length w. To show the result, one can perform a simple algorithm that finds a maximum matching in polynomial time: For each center node $v_A \in V_A$, one has to check whether there exists at least one connection to every peer V_{B_i} with $i \in \{1, \dots, N-1\}$. If so, one chooses one edge to each peer and adds all these nodes together with the center node as an N-tuple to the matching. In the case of the WQRMM, one chooses the connection with the highest weight. Since there is no other link from the peer node to the center, it is always better to choose this node instead of not including the node at all, and no better total weight can be reached. Result 3 covers the case that all connections are allowed, i.e., the connection length is maximal. In that case, we can immediately see the cardinality of the maximum quantum router matching since it is given by the minimum number of filled memories of one party. As the N-tuples, one chooses the first $|M|$ nodes from each partition $V_A, V_{B_1}, \dots, V_{B_{N-1}}$ and adds them to the matching. As this empties all memories of at least one partition, no more N-tuples can be added to the matching so that it is maximal. Similar to result 3, result 4 covers the setup in which the connection length is not restricted, this time including weights. The *rearrangement inequality* [Rud52] states that for three ordered sequences of nonnegative real numbers

$$x_1 \leq x_2 \leq \dots \leq x_n, \quad y_1 \leq y_2 \leq \dots \leq y_n, \quad z_1 \leq z_2 \leq \dots \leq z_n,$$

and the two permutations

$$y_{\pi(1)}, y_{\pi(2)}, \dots, y_{\pi(n)} \text{ of } y_1, \dots, y_n, \quad z_{\tau(1)}, z_{\tau(2)}, \dots, z_{\tau(n)} \text{ of } z_1, \dots, z_n$$

it holds:

$$x_1 y_{\pi(1)} z_{\tau(1)} + \dots + x_n y_{\pi(n)} z_{\tau(n)} \leq x_1 y_1 z_1 + \dots + x_n y_n z_n. \tag{3.2}$$

The same holds for any finite number of ordered sequences of nonnegative real numbers. The details of the proof of the rearrangement inequality are given in [Rud52]. Since all weights $W \in [0, 1]$ are nonnegative real numbers, we can use the rearrangement inequality to show that ordering all nodes by their weights and forming the N-tuples choosing the first $|M|$ nodes with highest weights of each partition always leads to a matching with maximum cardinality and maximum weight. The cardinality of the matching is again given by the minimum number of nodes of one partition.

For the tripartite network, every instance of the unweighted multipartite matching can be mapped to an instance of the well-known *Network Flow problem* (see for example [Wil19, AMO93]) since this setup always represents a flow from peer B_1 over the center party A to peer B_2. Accordingly, the 3-partite matching can always be solved efficiently via network flow. This holds for the full-range multiplexing (i.e., maximal connection length), as well as for the finite-range multiplexing (i.e., restricted connection length).

The underlying *maximum network flow* is defined as follows: Given a directed graph $G = (V, E)$ with two additional vertices $s, t \in V$. These are the *source* s from which the flow goes to the *sink* t. Each edge (v_1, v_2) is assigned to a capacity $u(v_1, v_2) \in \mathbb{R}_0^+$. The *s-t flow* has to fulfill the

a) *capacity constraint*: each edge $(v_1, v_2) \in E$ has a maximum flow upper bounded by the capacity $u(v_1, v_2)$,

b) *flow conservation constraint*: for each node $v_i \in V$, the sum of the input flow equals the sum of the output flow of that node v_i.

The goal of the network flow is to find a flow f that maximizes the net flow from the source s to the sink t.

To correctly model the 3-partite matching, all capacities are set to one. In this way, it is guaranteed that memories from each peer node B_1 and B_2 appear only once in a matching. To make the center node's memories appear only once, the array of nodes from party A has to be duplicated to A'. Edges between layers A and A' are only drawn between a node and its own copy. Since the capacities are restricted to one, and input and output flows have to be equal, it is now guaranteed that also the nodes from party A appear only once in a matching. Fig. 3.5 shows an example network flow problem for a given quantum router instance. As the general multipartite matching is defined over bipartite edges connecting the center party A to the peers B$_i$, it is generally not possible to map the multipartite matching (with $N > 3$) to the maximum network flow problem.

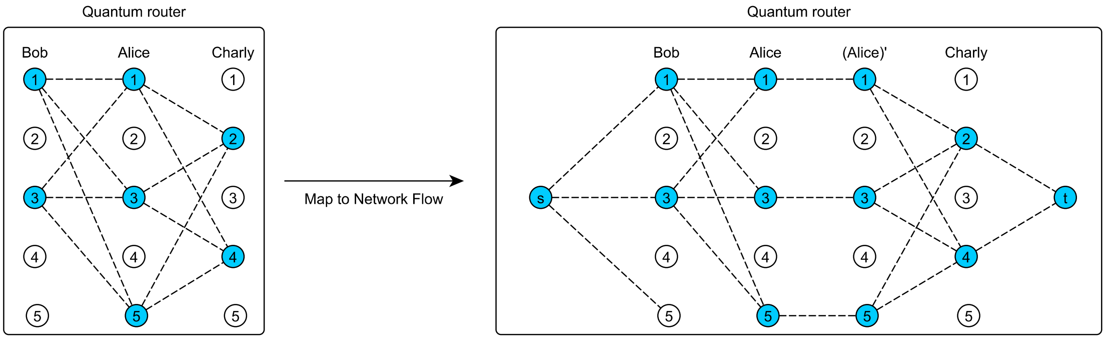


Figure 3.5: Example of a Network flow instance. On the left, the quantum router with its memories is shown. The right side shows the resulting network flow instance, with the additional source s and sink t. To prevent Alice's memories from being chosen twice, the copied layer (Alice)' is added, resulting in a single edge between a node from A and A'.

3.4.2 Approximation algorithm

For the general maximum quantum router matching in which we can restrict the connection length but do not include weights to the nodes, we provide an approximation algorithm. The general idea of this algorithm is to first include bipartite edges between the center node V_A and the peers V_{B_i} for nodes that only have one edge to the center or the peer, respectively. All these nodes are added to the matching, and further connections of these nodes are deleted. In this way, it is often possible to resolve the matching step by step. Algorithm 1 shows the working principle of the approximation algorithm. The cardinality of the matching can – for a restricted connection length – only be upper bounded by

$$|M| \leq \min_{\iota \in \{A, B_1, \ldots, B_{N-1}\}} |V_\iota|. \tag{3.3}$$

Note that in all subsets $V\iota$, only filled memories are considered when calculating the upper bound on $|M|$, as empty memories do not contribute to the matching. With the instruction *Remove orphaned nodes*, we mean that all center nodes that are not connected to at least one node from every peer B$_i$ and vice versa are deleted. The function *Get l_max* calculates the maximum cardinality as $|M| = \min_{\iota \in \{A, B_1, \ldots, B_{N-1}\}} |V_\iota|$.

Algorithm 1 Approximation Algorithm

```
remove orphaned nodes
get $l\_max$

                                        ▷ Strategy 1

repeat ← True
while repeat do
    repeat ← False
    check_1 ← True
    while check_1 do
        check_1 ← False
        for $v^j \in V_A$ do
            if $v^j$ has a single unmarked edge to any $v^{j'} \in V_{B_i}$ then
                mark edge $\{v^j, v^{j'}\}$ for later selection of matching
                delete further edges from node $v^{j'} \in V_{B\_i}$
                delete orphaned nodes
                check_1 ← True
                repeat ← True
            end
        end
    end
    if marked edges give valid matching with cardinality $l\_max$ then
        return matching M
    end
    check_2 ← True
    while check_2 do
        check_2 ← False
        for $v^{j'} \in V_{B_i}$ do
            if $v^{j'}$ has single unmarked edge to any node $v^j \in V_A$ then
                mark edge $\{v^j, v^{j'}\}$ for later selection of matching
                delete further edges from node $v^j \in V_A$
                delete orphaned nodes
                check_2 ← True
                repeat ← True
            end
        end
    end
    if marked edges give valid matching with cardinality $l\_max$ then
        return matching M
    end
end
for $v^j \in V_A$ do
    if $v_j$ has no marked edge to a node $v^{j'}$ in any $V_{B\_i}$ then
        choose first edge between center $V_A$ and peer $V_{B_i}$ and mark it
    end
end
if marked edges give valid matching with cardinality $l\_max$ then
    return matching M
end

                                        ▷ Strategy 2

reset graph (i.e., unmark all edges and add deleted nodes and edges)
for $v^j \in V_A$ do
    for $v^{j'} \in V_{B_i}$ do
        mark the first edge for matching M'
        delete orphaned nodes

    end
end
if $|M'| < |M|$ then
    return matching M
end
return matching M'
```

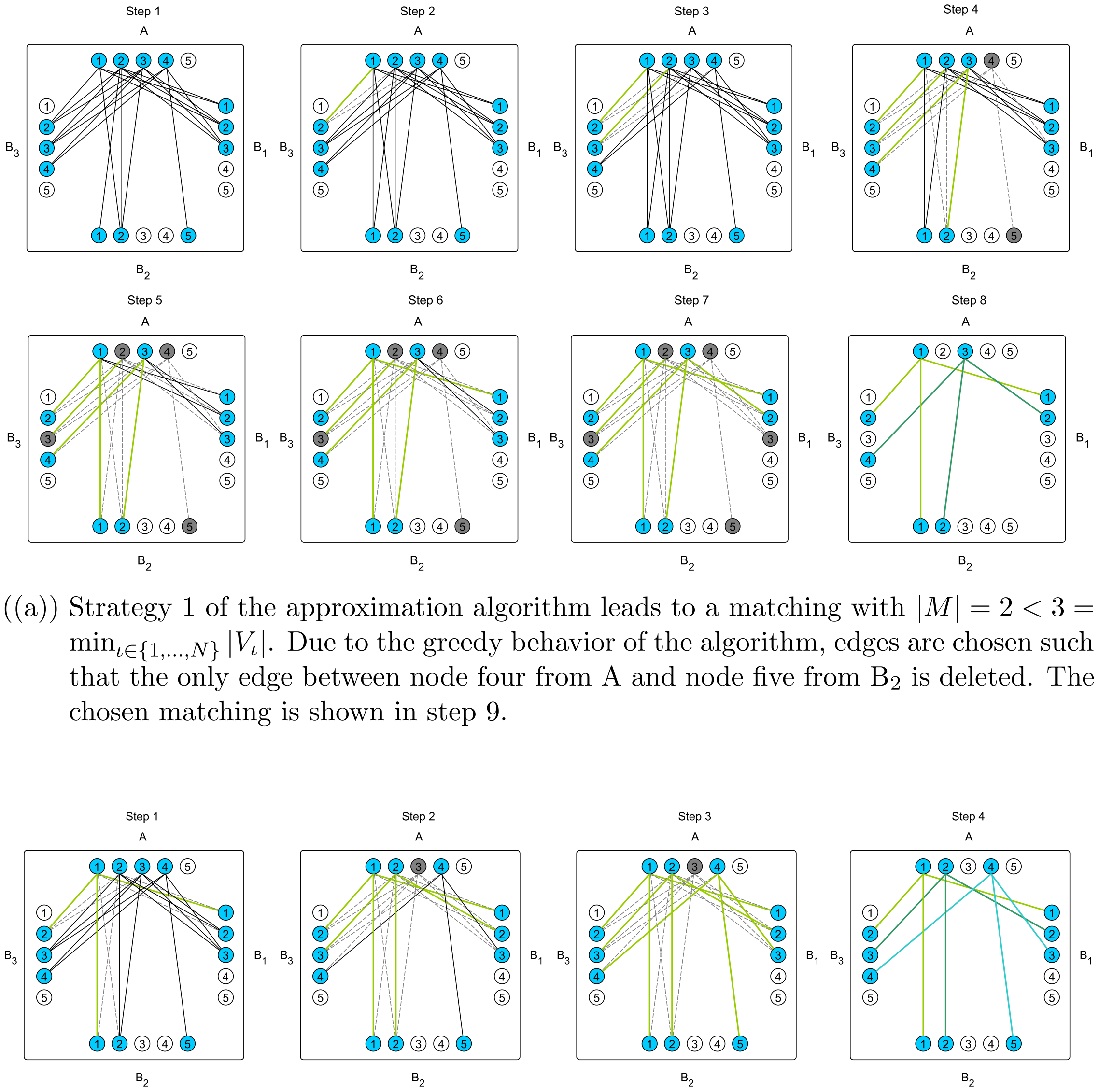
Given a valid graph instance representing the quantum router at a specific time, the algorithm either

- terminates with strategy 1 and finds a matching with maximum cardinality $|M| = \min_{\iota \in \{1,\dots,N\}} |V_\iota|$ or
- does not terminate with strategy 1, and a matching with cardinality $|M| < \min_{\iota \in \{1,\dots,N\}} |V_\iota|$ is found. This can happen for the following reasons:
  - since Eq. 3.3 is only an upper bound, it is possible that the matching has maximum cardinality with $|M| < \min_{\iota \in \{1,\dots,N\}} |V_\iota|$ or
  - due to the greedy behavior of the algorithm, edges may not be considered that are generally necessary to reach a maximum cardinality.

  In this case, strategy 2 is performed instead. Since edges are randomly chosen here, the algorithm either terminates with a matching that has maximum cardinality or with any matching M'. If $|M| > |M'|$, i.e., Strategy 1 leads to a matching with larger cardinality, this matching is returned instead.

An example input graph on which the algorithm does not terminate with Strategy 1 due to its greedy behavior is shown in Fig. 3.6. Figure 3.6(a) shows how the greedy algorithm performs in Strategy 1. The resulting matching has a cardinality smaller than 1_max since edges are deleted by the algorithm that are needed to get a maximum cardinality matching. In Fig. 3.6(b), the choice given by Strategy 2 of the greedy algorithm is shown. In this example, a matching with maximum cardinality is achieved. Note that the matching that is found by the algorithm depends on the order in which the for-loops go through the graph. Starting in reverse order would make strategy 1 find the maximum cardinality matching in this example, while strategy 2 would not lead to a maximum cardinality matching.

In Fig. 3.7, we give an overview of how the approximation algorithm performs. Therefore, we run the protocol over 500 rounds and repeat that 100 times to get a total of 50,000 samples. In each round, we track whether there is no matching, a matching with only one N-tuple (always choose the first N-tuple), or whether it terminates with Strategy 1 or 2. By definition, the algorithm always finds a



((a)) Strategy 1 of the approximation algorithm leads to a matching with $|M| = 2 < 3 = \min_{\iota\in\{1,\ldots,N\}} |V_\iota|$. Due to the greedy behavior of the algorithm, edges are chosen such that the only edge between node four from A and node five from B_2 is deleted. The chosen matching is shown in step 9.

((b)) Strategy 2 of the approximation algorithm results in a matching with maximum cardinality $|M| = \min_{\iota\in\{1,\ldots,N\}} |V_\iota| = 3$. The chosen matching is shown in step 4.

Figure 3.6: Approximation algorithm applied to an example graph instance representing a quantum router with four parties and five memories each. The connection length is restricted to $w = 1$.

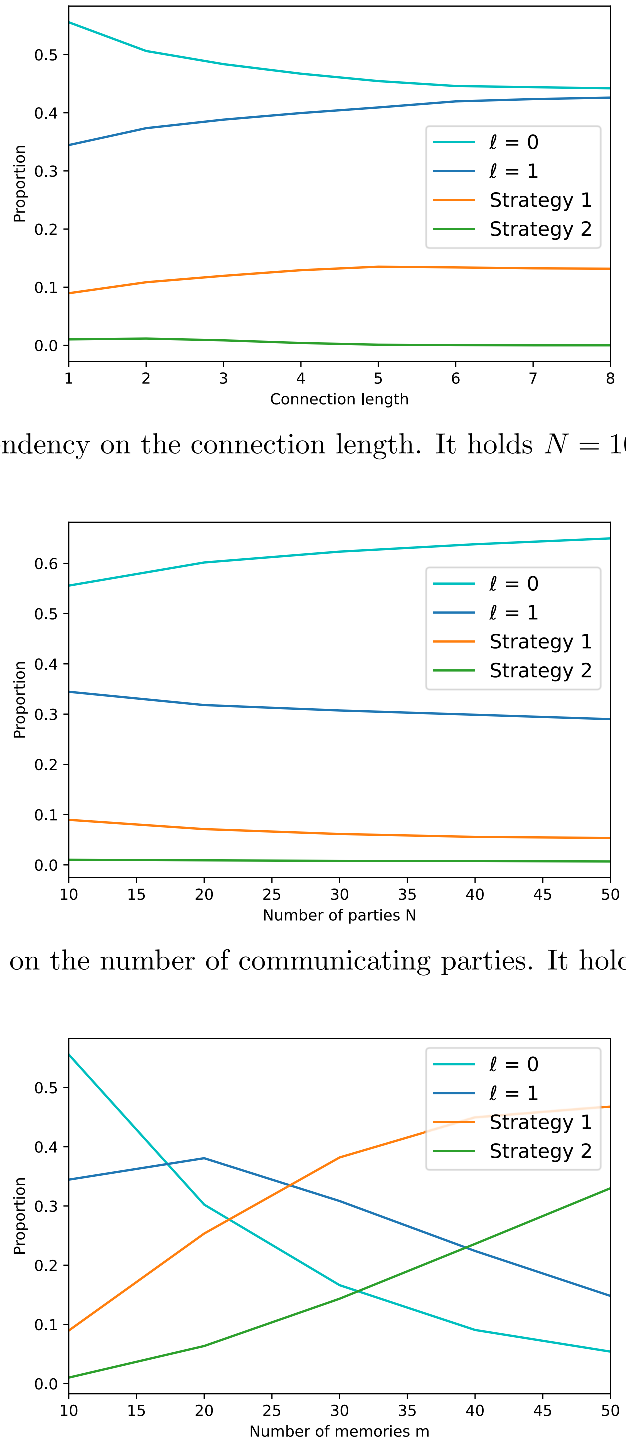
maximum matching with cardinality l_max when it terminates with strategy 1. For Strategy 2, we cannot say whether it finds a maximum matching by chance.

For small network sizes, we additionally run an exact algorithm that finds a maximum cardinality matching from all possible solutions (see Algorithm 2). Therefore,

Algorithm 2 Exact Algorithm

```
get l_max
for $v^j \in V_A$ do
    for ${v^j}' \in V_{B_i}$ do
        mark all edges $\{v^j, {v^j}'\}$ to all peers $B_i$
    end
    hyperedges_$v^j$ ← combinations(marked_edges, N-1)
    for hyperedge in hyperedges_$v^j$ do
        delete hyperedge, if there is more than one edge to the same $B_i$
    end
end
l ← l_max
while $l > 0$ do
    combination_of_hyperedges ← combinations(hyperedges, l_bound)
    for hyperedges in combination_of_hyperedges do
        delete all hyperedges, where elements are not pairwise disjoint
        if len(hyperedges) = $l$ then
            return l
        end
    end
    $l = l - 1$
end
```

we first calculate all possible hyperedges for each of Alice's nodes $v^j \in V_A$. To do this, we first consider all possible combinations of bipartite edges of a node $v^j \in V_A$ and select from these combinations all of them that have exactly one edge to each peer B_i. In the example given in Fig. 3.8, the hyperedges $\{v_A^1, v_{B_1}^1, v_{B_1}^2\}, \{v_A^1, v_{B_1}^1, v_{B_2}^2\}$, and $\{v_A^1, v_{B_1}^2, v_{B_2}^2\}$ are selected for node v_A^1. However, since the hyperedge $\{v_A^1, v_{B_1}^1, v_{B_1}^2\}$ does not contain a node from party B_2, it is invalid and therefore deleted. Valid hyperedges inclduing node v_A^1 are then given by $\{v_A^1, v_{B_1}^1, v_{B_2}^2\}$ and $\{v_A^1, v_{B_1}^2, v_{B_2}^2\}$. The same is done for all other nodes $v^j \in V^A$.



((a)) Dependency on the connection length. It holds $N = 10$ and $m = 10$.

((b)) Dependency on the number of communicating parties. It holds $m = 10$ and $w = 1$.

((c)) Dependency on the number of memories per party. It holds $N = 10$ and $w = 1$.

Figure 3.7: Dependency of the greedy algorithm performance on different network sizes and connection lengths. In all cases, the probability of link generation is set to $p = 0.1$.

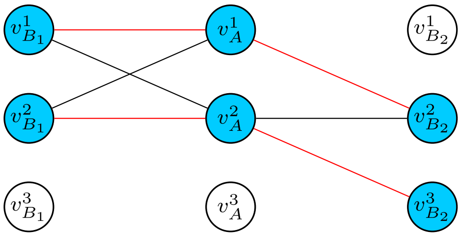


Figure 3.8: Example graph instance for the exact algorithm to determine a valid matching. The valid matching is shown in red.

Given all valid hyperedges, we calculate all possible combinations of these hyperedges from which we want to find a valid solution, i.e., one that does not include the same node in more than one hyperedge. In the given example, a valid combination is given by $M = \{\{v_A^1, v_{B_1}^1, v_{B_2}^2\}, \{v_A^2, v_{B_1}^2, v_{B_2}^3\}\}$ since this fulfills all requirements given in Sec. 3.3. To get all combinations of a given set, we use Python's *itertools.combinations* module (referred to as "combinations" in the algorithm) [Pyt24]. Nevertheless, due to the exponential runtime of the multipartite matching, the exact algorithm works with our hardware only up to a network size of five parties with five memories each. For this network size, we find that for a success probability (for generating a bipartite link between the peers and the central station) of $p = 0.1$ and all restricted connection lengths, there is a matching with cardinality $|M| \geq 1$ in about 30% of the rounds, and the approximation algorithm always finds a maximum cardinality matching. For a success probability of $p = 0.2$ and a connection length of $w = 1$, with a probability of under 0.1% no maximum matching is found. Additionally, for arbitrary connections between the filled nodes, the approximation algorithm fails to yield a maximum matching in under 1% of the cases for a success probability of $p = 0.1$.

For larger network sizes, we give the statistics of how often the approximation algorithm terminates with $|M| = 0$, $|M| = 1$, strategy 1, or strategy 2 in Fig. 3.7. For the analysis, we fix the success probability of generating a bipartite link between the parties and the central router to $p = 0.1$. According to Eq. 2.53, a success probability of 10% corresponds to a distance between the peers and the central node of $d = 50$ km. Smaller distances, i.e., larger probabilities, are not realistic for real-world applications, and smaller probabilities, i.e., larger distances, lead to very low key rates and are, therefore, also not feasible. In Fig. 3.7(a), the dependency on different connection lengths $w \in [1, 8]$ is shown for a fixed network size of $N = 10$ and $m = 10$. It turns out that strategy 2 is only relevant for small w. Therefore, we set this parameter to $w = 1$ for further analyses. An increase in the number of communicating parties also does not influence the result of the algorithm remarkably, as shown in Fig. 3.7(b). For a fixed number of $m = 10$, the algorithm terminates with strategy 2 in at most 1% of the cases. With increasing N, the percentage drops further below 1%.

Fig. 3.7(c) shows that only with increasing m, the algorithm terminates significantly often with strategy 2. For the considered setup of $N = 10$ and $w = 1$, the algorithm terminates with strategy 2 in more than 30% of the rounds. Compared to the setup with $N = 10$ and $m = 10$, where only 1% of the rounds terminate with strategy 2, this is an increase of about 3000%. This shows that the algorithm works only approximately and is less accurate for larger network sizes regarding the number of memories per party. Nevertheless, we have to take into account that the completion with strategy 2 does not necessarily mean that the algorithm did not find a matching with maximum cardinality.

3.5 Conclusion

In our work, we properly define the graph-theoretical problem of quantum router matching needed to optimally choose the qubits for the GHZ measurements in multipartite quantum routers. We show the difference to the known N-dimensional matching problem and study its computational complexity. Although the general decision problem is proven to be $\mathcal{NP}$-complete, i.e., intractable, we show that in some exceptions, the problem is efficiently solvable, i.e., there exist efficient algorithms. With these algorithms, it is possible to simulate multipartite quantum routers in a feasible time and make clear how the memories have to be entangled via a GHZ state measurement to increase the router rate while decreasing the storage times of the individual qubits, thus reducing the QBERs.

With the approximation algorithm, we show that it is already possible to deal with the general unweighted maximum multipartite matching, even though it does not always lead to a matching with maximum cardinality. With our simulations, we show that for smaller network sizes, the algorithm finds a maximum cardinality matching in over 90% of the cases. Only for increasing m does the algorithm become less exact in more rounds. In future work, we propose to focus further on the approximation algorithm by considering the order of the for-loops. Starting to check for single edges at different nodes can influence the algorithm's result (as shown by the example given in Fig. 3.6). We further propose to include weights in the approximation algorithm since the weighted quantum router maximum matching is the more interesting case from the physical point of view.

3.6 Publication

The complexity proof together with the algorithms solving the special cases we consider here are accepted as a contribution at the *2nd Workshop on Quantum Algorithms, Software, and Applied Research* (QUASAR 2025) and will be published in the Proceedings of this conference [BGK$^+$25]. The accepted version of the paper is attached to this work in Appendix A. A second paper, including the results of the approximation algorithm, will be submitted to the Journal of *Advanced Quantum Technologies*, and is currently in preparation [JAK25].

3.7 Personal contribution

The $\mathcal{NP}$-completeness proof of the general quantum router matching was done by Christian Laußmann. The results 1-3 about special cases being in $\mathcal{P}$ were done by Christian Laußmann and me in equal parts. Result 4, which was about the weighted quantum router maximum matching with unrestricted connection length, was done by Luis Gindorf. The approximation algorithm and the exact algorithm were developed and tested in simulations by me. Dorothea Baumeister came up with the idea of using network flow for the tripartite quantum router matching, which was then implemented and analyzed by me. The accepted publication was written by Luis Gindorf, Christian Laußmann, Jörg Rothe, and me in equal parts.

CHAPTER 4

EFFICIENT MULTIPLEXING STRATEGIES IN QUANTUM ROUTERS

4.1 Introduction

In this chapter, we generalize the concept of the quantum repeater to the multipartite setup of a quantum router. In contrast to [LFL$^+$23b], we use quantum memories to improve the key rate through multiplexing. In our work, we investigate in detail how the use of such quantum memories affects the qubits and the achievable rates in quantum key distribution. On the one hand, memory multiplexing helps increase the router rate, i.e., multiplexing leads to an increase in the number of distributed GHZ states per round. On the other hand, however, storing the qubits in the memories leads to an interaction between the qubits and their environment. This means the qubits lose their information over time, and their fidelity decreases. The longer a qubit is stored in memory, the more likely it is to interact with its surroundings, which can cause decoherence. As a result, the quantum bit error rates increase, and the fraction of bits that can be used to exchange the secret key becomes smaller.

In [AKB14a], Abruzzo et al. investigate different matching strategies for the bipartite quantum repeater. In our work, we first extend the principles of a quantum repeater using memory multiplexing to the multipartite setup of a quantum router being placed between N parties. In a second step, we deal with the question of how to optimally use the memories and connect qubits via a GHZ state measurement without the QBERs becoming too large. For this investigation, we use the implementations of the (weighted) quantum router maximum matching introduced in Sec. 3.

Before we consider the matching strategies, we generalize the setup and adapt the formulas needed to calculate the secret key rate in the (multipartite) N-BB84 protocol.

4.2 Conference key agreement via quantum routers

We consider a star graph with one central quantum router and N parties with m memories each located around it as depicted in Fig. 2.11. To distribute multipartite entangled states among all N parties, each party generates bipartite Bell states between itself and the central quantum router first. With a GHZ state measurement, the locally held qubits are projected onto one of the GHZ states. By applying local operations depending on the measurement outcome, the desired $\left|GHZ_0^+\right\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right)$ can be produced.

Once stored in the memories, the qubits start to decohere. Here, we restrict the noise model to depolarization (see Chapter 8.3.4 in [NC10], for example). A qubit that is stored for δ rounds in the memory is then given by:

$$\rho(\delta) = p(\delta)\rho_0 + \frac{1-p(\delta)}{2}\mathbb{1}, \tag{4.1}$$

with initial state ρ_0. With probability $p(\delta)$, the state remains untouched, and with probability $1-p(\delta)$, it is replaced by the maximally mixed state $\mathbb{1}/2$. The probability of white noise relates to the decoherence parameter τ of the quantum memory via

$$p(\delta) = e^{-\delta/\tau}. \tag{4.2}$$

If the qubit is not stored, i.e., for $\delta = 0$, we find $p(\delta = 0) = 1$, so that the qubit remains untouched. For $\delta > 0$, the qubit decays exponentially. The decoherence parameter τ defines the number of storage rounds after which a quantum state decays to $1/e$ of its initial value, meaning it loses about 63% of its coherence. The bipartite states, after being stored in the memory, are described by a Werner-like state with fidelity F:

$$\rho = F \left|\phi^+\right\rangle \left\langle\phi^+\right| + \frac{1-F}{3} \left(\left|\phi^-\right\rangle \left\langle\phi^-\right| + \left|\psi^+\right\rangle \left\langle\psi^+\right| + \left|\psi^-\right\rangle \left\langle\psi^-\right|\right). \tag{4.3}$$

Due to the storage, the fidelity of state ρ changes according to

$$F \rightarrow \frac{1}{4} + \left(F - \frac{1}{4}\right) e^{-\delta/\tau}. \tag{4.4}$$

The total initial state on which the GHZ state measurement is performed is given by the tensor product of all $\iota \in \{1, 2, \ldots, N\}$ parties' initial Bell diagonal states ρ_ι, each with fidelity F_ι:

$$\begin{aligned}\rho_{tot} &= \otimes_{\iota=1}^{N} \rho_\iota \\ &= \otimes_{\iota=1}^{N} \left(F_\iota \left|\phi^+\right\rangle \left\langle\phi^+\right| + \frac{1-F_\iota}{3} \left(\left|\phi^-\right\rangle \left\langle\phi^-\right| + \left|\psi^+\right\rangle \left\langle\psi^+\right| + \left|\psi^-\right\rangle \left\langle\psi^-\right|\right)\right). \end{aligned} \tag{4.5}$$

Performing the GHZ state measurement and local operations as given in Sec. 2.2.2 and tracing out the qubits in the central station leads to a GHZ diagonal state held locally by the N parties. This state is of the following form:

$$\begin{aligned}\rho^{GHZ} =& \lambda_0^+ \left|GHZ_0^+\right\rangle \left\langle GHZ_0^+\right| + \lambda_0^- \left|GHZ_0^-\right\rangle \left\langle GHZ_0^-\right| \\ &+ \sum_{\kappa=1}^{\frac{2^N-2}{2}} \lambda_\kappa \left(\left|GHZ_\kappa^+\right\rangle \left\langle GHZ_\kappa^+\right| + \left|GHZ_\kappa^-\right\rangle \left\langle GHZ_\kappa^-\right|\right), \end{aligned} \tag{4.6}$$

with the general GHZ states as given in Eq. 2.51. Although the state is GHZ diagonal, we consider the density matrix expressed in the computational basis in the following. Note that due to symmetry, the second half of the diagonal elements equals the first half in reverse order. Equality also holds for the two off-diagonal elements $\rho^{GHZ}_{(1,2^N)}$(upper right corner) and $\rho^{GHZ}_{(2^N,1)}$ (lower left corner). The GHZ diagonal elements λ_κ can be calculated explicitly assuming that the qubits undergo depolarization during storage. For an explicit calculation, we rewrite Eq. 4.3 as

$$\rho = A\left|\phi^+\right\rangle\left\langle\phi^+\right| + B\left|\phi^-\right\rangle\left\langle\phi^-\right| + C\left|\psi^+\right\rangle\left\langle\psi^+\right| + D\left|\psi^-\right\rangle\left\langle\psi^-\right| \tag{4.7}$$

with Bell diagonal elements $A = F$ and $B = C = D = \frac{1-F}{3}$. The elements of the initial state ρ for each party $\iota \in \{1, 2, \ldots, N\}$ are given by

$$\rho_{(1,1),\iota} = \rho_{(4,4),\iota} = \frac{A+B}{2} = \frac{F_\iota}{3} + \frac{1}{6}, \tag{4.8}$$

$$\rho_{(1,4),\iota} = \rho_{(4,1),\iota} = \frac{A-B}{2} = \frac{2}{3}F_\iota - \frac{1}{6}, \text{ and} \tag{4.9}$$

$$\rho_{(2,2),\iota} = \rho_{(3,3),\iota} = \frac{C+D}{2} = \frac{1}{3} - \frac{F_\iota}{3}. \tag{4.10}$$

With these equations, the elements of the shared GHZ state ρ^{GHZ} can be calculated. For the elements $\rho^{GHZ}_{(1,1)} = \rho^{GHZ}_{(2^N,2^N)}$ and $\rho^{GHZ}_{(1,2^N)} = \rho^{GHZ}_{(2^N,1)}$ we find:

$$\rho^{GHZ}_{(1,1)} = \chi_{norm}\frac{\lambda_0^+ + \lambda_0^-}{2} = \prod_{\iota=1}^{N}\left(\rho_{(1,1),\iota} + \rho_{(2,2),\iota}\right) = \rho^{GHZ}_{(2^N,2^N)} \tag{4.11}$$

$$\rho^{GHZ}_{(2^N,1)} = \chi_{norm}\frac{\lambda_0^+ - \lambda_0^-}{2} = \prod_{\iota=1}^{N}\rho_{(4,1),\iota} = \rho^{GHZ}_{(1,2^N)} \tag{4.12}$$

with normalization constant χ_{norm}. All other entries along the diagonal in ρ^{GHZ} are of the form

$$\chi_{norm}\left(\rho_{(1,1),1}\prod_{i=2}^{N}\rho_{(x,x),i} + \rho_{(2,2),1}\prod_{i=2}^{N}\rho_{(\bar{x},\bar{x}),i}\right) \tag{4.13}$$

with indices (x, x) either being entry $(1, 1)$ or entry $(3, 3)$ and the indices $(\bar{x}, \bar{x})$ indicate the negation of (x, x), i.e., they are given by $(2, 2)$ or $(4, 4)$. For each diagonal entry in ρ^{GHZ}, the indices (x, x) are chosen such that all possible combinations of indices appear once in the matrix. The normalization constant guarantees, that $tr\left(\rho^{GHZ}\right) = 1$. It is given by $\chi_{norm} = 2^{N-1}$.

For the tripartite network with parties A, B, and C, for example, we find the following diagonal matrix entries:

$$\begin{aligned}
\rho^{GHZ}_{(1,1)} &= \rho^{GHZ}_{(8,8)} = \frac{\lambda_0 + \lambda_0}{2} \\
&= 2^2 \left(\rho_{(1,1),A}\rho_{(1,1),B}\rho_{(1,1),C} + \rho_{(2,2),A}\rho_{(2,2),B}\rho_{(2,2),C}\right), && (4.14) \\
\rho^{GHZ}_{(2,2)} &= \rho^{GHZ}_{(7,7)} = \lambda_1 \\
&= 2^2 \left(\rho_{(1,1),A}\rho_{(1,1),B}\rho_{(3,3),C} + \rho_{(2,2),A}\rho_{(2,2),B}\rho_{(4,4),C}\right), && (4.15) \\
\rho^{GHZ}_{(3,3)} &= \rho^{GHZ}_{(6,6)} = \lambda_2 \\
&= 2^2 \left(\rho_{(1,1),A}\rho_{(3,3),B}\rho_{(1,1),C} + \rho_{(2,2),A}\rho_{(4,4),B}\rho_{(2,2),C}\right), \text{ and} && (4.16) \\
\rho^{GHZ}_{(4,4)} &= \rho^{GHZ}_{(5,5)} = \lambda_3 \\
&= 2^2 \left(\rho_{(1,1),A}\rho_{(3,3),B}\rho_{(3,3),C} + \rho_{(2,2),A}\rho_{(4,4),B}\rho_{(4,4),C}\right). && (4.17)
\end{aligned}$$

For the off-diagonal matrix elements, it holds

$$\begin{aligned}
\rho^{GHZ}_{(1,8)} &= \rho^{GHZ}_{(8,1)} = \frac{\lambda_0^+ - \lambda_0^-}{2} \\
&= 2^2 \left(\rho_{(4,1),A}\rho_{(4,1),B}\rho_{(4,1),C}\right). && (4.18)
\end{aligned}$$

Generally, the GHZ diagonal elements λ_κ with $\kappa \in \{1, \ldots, \frac{2^N-2}{2}\}$ directly follow from Eq. (4.13) , while the elements λ_0^+ and λ_0^- are calculated from Eq. (4.11) and Eq. (4.12):

$$\begin{aligned}
\lambda_0^+ &= \chi_{norm} \left(\prod_{\iota=1}^{N} \rho_{(1,1),\iota} + \prod_{\iota=1}^{N} \rho_{(2,2),\iota} + \prod_{\iota=1}^{N} \rho_{(4,1),\iota} \right) \\
&= \chi_{norm} \left(\prod_{\iota=1}^{N} \left(\frac{F_\iota}{3} + \frac{1}{6}\right) + \prod_{\iota=1}^{N} \left(\frac{1}{3} - \frac{F_\iota}{3}\right) + \prod_{\iota=1}^{N} \left(\frac{2}{3}F_\iota - \frac{1}{6}\right) \right) && (4.19) \\
\lambda_0^- &= \chi_{norm} \left(\prod_{\iota=1}^{N} \rho_{(1,1),\iota} + \prod_{\iota=1}^{N} \rho_{(2,2),\iota} - \prod_{\iota=1}^{N} \rho_{(4,1),\iota} \right) \\
&= \chi_{norm} \left(\prod_{\iota=1}^{N} \left(\frac{F_\iota}{3} + \frac{1}{6}\right) + \prod_{\iota=1}^{N} \left(\frac{1}{3} - \frac{F_\iota}{3}\right) - \prod_{\iota=1}^{N} \left(\frac{2}{3}F_\iota - \frac{1}{6}\right) \right). && (4.20)
\end{aligned}$$

Note that the GHZ diagonal element λ_0^+ of the new state ρ^{GHZ} also gives the fidelity of the state, i.e., $F^{GHZ} = \left\langle GHZ_0^+ \right| \rho^{GHZ} \left| GHZ_0^+ \right\rangle = \lambda_0^+$.

Regarding the conference key agreement (for details about the N-BB84 protocol, see Sec. 2.4), we find for the QBERs the following expressions:

$$\begin{aligned}
Q_X &= \frac{1-\left(\lambda_0^+ + \lambda_0^-\right)}{2} \\
&= \frac{1}{2} - \chi_{norm} \prod_{\iota=1}^{N} \left(\frac{2}{3}F_\iota - \frac{1}{6}\right) \\
&= \frac{1}{2} - \frac{1}{2\cdot 3^N} \prod_{\iota=1}^{N} \left(4F_\iota - 1\right) \qquad (4.21) \\
Q_{AB_i} &= \frac{1-\langle Z_A Z_{B_i}\rangle}{2} \\
&= 2\chi_{norm} \left(\frac{1}{2}\right)^{N-2} \left(\left(\frac{F_A}{3}+\frac{1}{6}\right)\left(\frac{1}{3}-\frac{F_{B_i}}{3}\right) + \left(\frac{F_{B_i}}{3}+\frac{1}{6}\right)\left(\frac{1}{3}-\frac{F_A}{3}\right)\right) \\
&= \frac{2}{9}\left(F_A + F_{B_i} - 4F_AF_{B_i} + 2\right). \qquad (4.22)
\end{aligned}$$

For the bipartite error rate, the probability of $Z_A \neq Z_{B_i}$ with centerparty A and peers B_i with $i \in \{1, 2, \ldots, N-1\}$ is fulfilled by all matrix entries containing either the elements $\rho^{GHZ}_{(1,1),A}\rho^{GHZ}_{(2,2),B_i}$ or $\rho^{GHZ}_{(1,1),B_i}\rho^{GHZ}_{(2,2),A}$; i.e., these elements contribute to Q_{AB_i}. All other terms with fidelities F_{B_j} with $B_i \neq B_j$ cancel out, leading to a constant prefactor of $(1/2)^{N-2}$. Due to the symmetry of ρ^{GHZ}, each term appears twice, thus giving the factor 2 in Eq. (4.22).

Since we perform the protocol over s rounds, the qubits used for a GHZ measurement in each round can experience different numbers of storage rounds within the memories. Therefore, we calculate the QBERs over all possible storage rounds $\delta_\iota \in \{0, 1, \ldots, s\}$ of qubits from all parties $\iota \in \{A, B_1, B_2, \ldots, B_{N-1}\}$:

$$\begin{aligned}
\bar{Q}_X(s) = \sum_{\delta_A=0}^{s} \sum_{\delta_{B_1}=0}^{s} \cdots \sum_{\delta_{B_{N-1}}=0}^{s} & Q_X(\delta_A, \delta_{B_1}, \ldots) \cdot Prob[\delta_A](s) \\
& Prob[\delta_{B_1}](s) \ldots Prob[\delta_{B_{N-1}}](s) \qquad (4.23)
\end{aligned}$$

$$\bar{Q}_{AB_i}(s) = \sum_{\delta_A=0}^{s} \sum_{\delta_{B_1}=0}^{s} \cdots \sum_{\delta_{B_{N-1}}=0}^{s} Q_{AB_i}(\delta_A, \delta_{B_1}, \dots) \cdot Prob[\delta_A](s)$$
$$Prob[\delta_{B_1}](s) \dots Prob[\delta_{B_{N-1}}](s). \tag{4.24}$$

This is the probability that a qubit with a certain number of storage rounds ($Prob(\delta_\iota)$) occurs multiplied by the QBER (see Eq. 4.21 for Q_X and Eq. 4.22 for Q_{AB_i}, respectively) resulting from that number of storage rounds. The total QBERs estimated during the protocol are given by the average QBER per round multiplied by the total number of successful measurements per round divided by the total number of measurements summed over the whole protocol up to a current round s_c:

$$Q_X^{tot} = \frac{\sum_{s=1}^{s_c} \langle \ell \rangle (s) \cdot \bar{Q}_X(s)}{\sum_{s=1}^{s_c} \langle \ell \rangle (s)} \tag{4.25}$$

$$Q_{AB_i}^{tot} = \frac{\sum_{s=1}^{s_c} \langle \ell \rangle (s) \cdot \bar{Q}_{AB_i}(s)}{\sum_{s=1}^{s_c} \langle \ell \rangle (s)}. \tag{4.26}$$

The number of measurements per round can be calculated similarly to the bipartite setup, as described in Sec. 2.3.1. For larger network sizes (exceeding a total number of 15 memories, i.e., $Nm > 15$), the average number of GHZ measurements per round is obtained via simulations. The same holds for the different numbers of storage rounds δ and the probabilities of how often these δ appear. An overview of the simulations is given in Sec. 4.3.

To calculate the asymptotic secret fraction, the total QBERs are inserted into Eq. 2.74:

$$r_\infty = 1 - h\left(Q_X^{tot}\right) - \max_{1 \leq i \leq N-1} h\left(Q_{AB_i}^{tot}\right). \tag{4.27}$$

Note that the asymptotic secret fraction can, in principle, become negative. From a physical perspective, a negative fraction of usable secret bits does not make sense. So, the value is then set to zero. Together with the router rate R (that is discussed in Sec. 2.3.2), the secret key rate can be calculated as the product of both these values, i.e.,

$$K = Rr_\infty. \tag{4.28}$$

4.3 Simulation

To calculate the router rate and the QBERs, the whole protocol of a quantum router with memory multiplexing is simulated. All simulations are done in *Python*. In the following, we give an overview of these simulations.

The quantum router with memory multiplexing

The quantum memories of each party are given by an array that can take either values of -1 (for empty memories) or a real number $F \in [0, 1]$, depending on the fidelity the qubit has according to its storage time. The steps of the simulations are as follows:

1. Starting from the empty memory configuration, a decision is made for each memory whether it is to be filled or not. That happens randomly with probability p. If a qubit is stored, the corresponding entry in the array is set to the initial fidelity F_{init} a qubit has. In the following rounds, only empty memories are tried to be filled.

2. At the beginning of each round, the fidelities of the leftover qubits from the previous round are reduced according to Eq. (4.4).

3. Based on the memory configuration, the (weighted) quantum router maximum matching is performed to decide which memories are involved in which GHZ measurement.

4. All array entries representing a qubit used in a GHZ measurement are reset to -1.

5. When cutoffs are considered, all memories with a fidelity $F < F_{cutoff}$ are deleted. That means that all corresponding array entries are also set to -1.

The whole protocol is performed for a fixed number of rounds. Additionally, it is repeated for several iterations to increase the number of samples for the statistics.

The matching problem

The matching is realized in different ways depending on the network size (N and m), the connection length w, and the strategy (unweighted or weighted) that is to be implemented.

(i) The quantum router maximum matching for $N = 3$ parties without weights is modeled as a maximum flow from peer B_1 over the central party A to peer B_2 (see Sec. 3.4 for details). We implement the network flow in Python with the module *maximum_flow* in *networkx* [Net24b].

(ii) For larger network sizes in the unweighted case, the implementation depends on the connection length. For $w = 0$ and $w = m - 1$, the matching is always efficiently solvable. The former has only one solution, since every node of B_i has at most one edge to the central party A. Only if A has a connection to all B_i, this N-tuple is added to the matching. The latter is solved analogously to the algorithm mentioned in Sec. 3.4.1 by successively adding the first filled memory from each party as an N-tuple.

For restricted connection lengths, the matching can be solved exactly for network sizes up to about 25 nodes in total (i.e., $Nm \leq 25$). That means that the first solution with maximum cardinality is chosen from all possible solutions. For networks including more than 25 nodes, the approximation algorithm from Sec. 3.4.2 is used.

(iii) The weighted matching problem was initially considered for small networks of up to three parties with four memories each. For this setup, a solution with maximum weights was initially selected from all solutions with maximum cardinality. In a later work, we introduced and proved the algorithm from Sec. 3.4.1 for efficiently solving the weighted maximum matching for a maximum connection length. For restricted connection lengths, a solution with maximum cardinality and weighting is still selected from all possible solutions.

Router rate and QBER

All parameters needed to calculate the secret key rate can be taken from these simulations. The router rate follows immediately from the simulation step 3 given above. The cardinality of the chosen matching performed in each round gives the number of GHZ state measurements ℓ performed. By repeating the simulations, an average $\langle \ell \rangle$ can be taken from these samples. The router rate follows from Eq. (2.68) by summing all average numbers of GHZ state measurements performed in each round up to the current round s_c and normalizing by the number of memories m and rounds s_c:

$$R(s_c) = \frac{1}{s_c} \sum_{s=1}^{s_c} \frac{\langle \ell \rangle (s)}{m}. \tag{4.29}$$

It is also possible to analytically calculate the router rate by following the equations given in Sec. 2.3.2. However, this is only possible for networks up to a total number of about 15 nodes since the transition matrices of size $2^{Nm} \times 2^{Nm}$ have to be calculated.

The quantum bit error rates are calculated via the fidelities of the qubits used in a GHZ measurement. These are taken from the simulations in step 3 of the previously given protocol. Using Eq. (4.21) for Q_X and Eq. (4.22) for Q_{AB_i}, the QBERs after the GHZ measurement are calculated. The average QBER per round can be either generated directly from these samples or all possible QBERs are calculated and weighted with the probability that qubits with this fidelity appear in a GHZ measurement (see Eq. (4.23) and Eq. (4.24)). The QBERs of one matching are averaged over all GHZ measurements performed in one protocol round. Given the router rate and the QBERs, the other values can be calculated following the corresponding equations given in Sec. 4.2.

4.4 Results

In our work, we perform simulations on the quantum router connecting $N = 3$ parties. Analogously to the results in a previous paper [AKB14a], we show with our simulations that the main advantage of using multiplexing is already given for a connection length of $w = 1$. Allowing GHZ measurements between memories with greater distance in the label does not lead to significant advantages. This result, however, should be taken carefully since the labels given to the quantum memories within a quantum router are artificial, and it is not fully clear how memories within the quantum router would be arranged in an experimental setup. Nevertheless, it shows that an all-to-all connection between the memories of different parties is not necessary to take advantage of multiplexing. This facilitates an experimental setup, as the all-to-all connections require the implementation of a GHZ state measurement circuit between all memories of the parties. Therefore, considering "direct neighbors" with a small connection length is already a good improvement for the achievable rates and, at the same time, easier to implement.

Our work focuses on the different matching strategies that can be performed to increase the secret key rate. In a first step, we analyze various strategies for choosing memories for the GHZ state measurements with the aim of maximizing the secret key rate. Second, we introduce cutoffs to the fidelities to further optimize the secret key rate. For the matching strategies, we weight all allowed connections in the quantum router. In general, weights are given by the quality of the qubits stored in the quantum memories. The quality is given by the fidelity of each qubit according to Eq. 4.4. In [BGK+25], we show numerically that the fidelities should be maximized in order to minimize the quantum bit error rates (from Eq. (4.21) and Eq. (4.22)). This can be seen from Eq. 4.21 as well, which is minimal for a maximal product, i.e., for the term $\prod_{\iota=1}^{N}(4F_\iota - 1)$ being maximal. That is fulfilled for all $F_\iota = 1$. With decreasing fidelities, the QBER in the X-basis increases. To minimize the QBER in the Z-basis, Eq. 4.22 has to be minimized. Also here, a minimum QBER is reached for maximal fidelities, i.e., $F_A = 1$ and $F_{B_i} = 1$. With decreasing fidelities, the QBER increases again. So in total, both QBERs are minimal with fidelity $F_\iota = 1$ for all parties $\iota \in \{1, 2, \dots, N\}$, and increase with decreasing fidelities.

Regarding the number of storage rounds δ, which determines the fidelities of the qubits, a minimization must be performed. Following this optimization, the newest qubits with the highest fidelities are always chosen first, while qubits with low fidelities are always less likely to be chosen and remain in the memories. Therefore, an additional strategy that chooses qubits with low fidelities is also considered. For this purpose, we combine new qubits with the older qubits, i.e., the difference in fidelities between Alice's qubit and the peers B_i are considered as weights and are either maximized or minimized. In total, we compare the following strategies:

S0: Random choice: choose the first matching with maximum cardinality independent of its weights.

S1: Difference in qubit quality: weight the connections by the difference in the number of storage rounds of the qubits included in an N-tuple and

a. minimize over the sum of weights, or

b. maximize over the sum of weights.

S2: Maximum qubit quality: maximize the fidelity of all qubits included in an N-tuple by minimizing the number of storage rounds.

Fig. 4.1 shows a comparison of the different strategies for a tripartite network with four memories per party. The plot shows that maximizing the fidelity of the qubits included in a matching (strategy S2) leads to the highest secret key rate overall. For a few rounds only, strategy S1 a.) performs better than S0. Here, minimizing the difference in the qubit quality leads to higher key rates than randomly choosing a matching with maximum cardinality. In the long term, the opposite holds: it is better to randomly choose the matching (S0) than to minimize the difference in the number of storage rounds (strategy S1 a.)). In contrast, maximizing the difference in the qubit quality (strategy S1 b.)) always leads to the lowest secret key rate. With strategy S1 a.), qubits with the highest correlations are chosen for each N-tuple. Consequently, qubits that have already experienced decoherence over a longer time are selected less likely and, therefore, decohere even further. On the contrary, strategy S1 b.) selects qubits with a significant difference in quality. Thus, qubits with a higher decoherence time are also selected. However, the resulting N-tuples have the lowest correlations even when N-tuples with perfect correlations could be generated.

As the preferred strategy S2 also does not choose qubits with lower fidelities, we additionally introduce cutoffs to remove qubits with low fidelities. Qubits that decohere in the quantum memories for a certain time are removed from the memories, so that these memories can be refilled with new qubits in the following. Under the assumption that $Q_X = Q_{AB_i} \equiv Q$ and that all fidelities are equal (i..e., $F_1 = F_2 = \dots = F_N \equiv F$) we get a maximal QBER of $Q \leq 0.11$ that still leads to a positive secret fraction according to Eq. (4.27). With that and Eq (4.21) we find for the

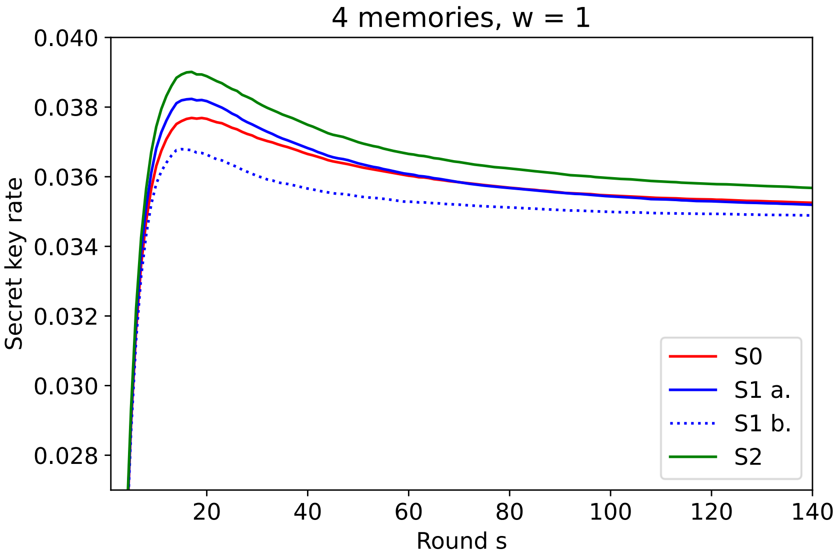


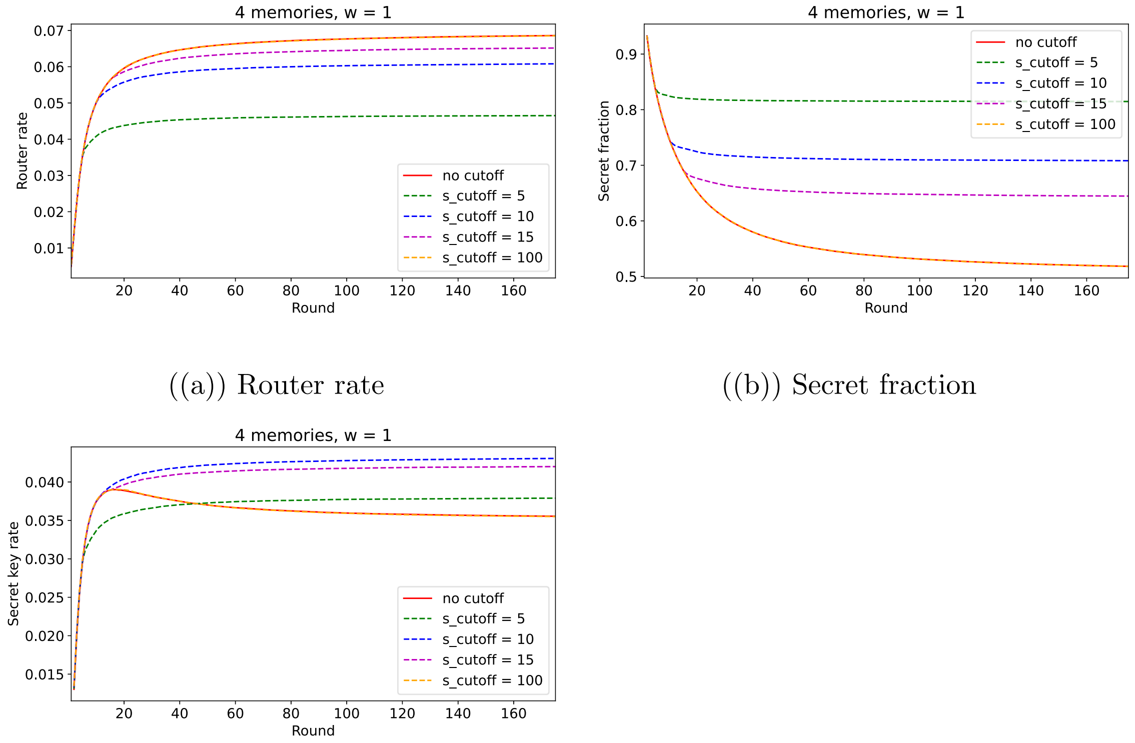
Figure 4.1: ([KKB24]) Comparison of the secret key rate for the different matching strategies in a tripartite network with four memories per party. The other parameters are: $w = 1$, $p = 0.1$, $\tau = 100$, and 50,000 samples.

QBER:

$$\begin{aligned} Q &= \frac{1}{2} - \frac{1}{2 \cdot 3^N}(4F-1)^N \leq 0.11 \\ \Leftrightarrow \frac{\sqrt[N]{(1/2-0.11) \cdot 2 \cdot 3 + 1}}{4} &\leq F. \end{aligned} \tag{4.30}$$

For the given setup of the tripartite network, the fidelity has to be $F \geq 0.94$, which is reached after about 8 storage rounds under the assumption that the initial fidelity is given by $F_{init} = 1$.

Since it holds in reality that $Q_X \leq Q_{AB_i}$ and not all fidelities are necessarily equal, the actual number of rounds that a qubit can be stored until the secret fraction becomes zero differs from that calculation. For the tripartite network considered here, the optimal cutoff time is 10 rounds. That follows from the simulation results shown in Fig. 4.2. To optimize the secret key rate, it must also be taken into account that by shortening the storage time, the router rate decreases. While maximizing the secret fraction, the router rate is reduced due to fewer GHZ measurements being performed every round. The optimal secret key rate is achieved when both effects



((a)) Router rate

((b)) Secret fraction

((c)) Secret key rate

Figure 4.2: ([KKB24]) Effect of the number of cutoff rounds on the rates in a tripartite network with four memories per party. The other parameters are: $w = 1$, $p = 0.1$, $\tau = 100$, and 50,000 samples.

cancel each other out, and the secret key rate does not decrease further with the number of rounds performed. For the tripartite network with $m = 4$ memories each, the effect on the router rate, the secret fraction, and the secret key rate are shown in Figure 4.2 (a)-(c) for various cutoff rounds. The plot of the secret key rate without cutoff (compare Fig. 4.2(c)) shows the competing behavior between the router rate and the secret fraction. For a small number of rounds, the secret key rate increases due to the increase in the router rate (see Fig. 4.2 (a)) while for a large number of rounds, the decrease in the secret fraction (see Fig. 4.2 (b)) predominates, such that the secret key rate decreases again. When choosing the right cutoff, these effects cancel out, and the secret key rate remains constant with the number of rounds. As seen in Fig. 4.2(c), this happens with a chosen cutoff of 10 rounds for the given network size. In general, the ideal cutoff can approximately be determined using Eq. 4.30

4.5 Conclusion

With this work, we generalize the bipartite quantum repeater with memories used for multiplexing to the multipartite quantum router. This setup can be seen as a generalization of the measurement-device-independent QKD protocol with quantum memories from [AKB14b] to the multipartite scenario. In comparison to previous work about conference key agreement in star graphs [LFL$^+$23b], we include quantum memories to analyze their effects and the advantage of multiplexing.

We show that the use of memory multiplexing increases the secret key rate. Similar to the bipartite quantum repeater, the main advantage is already gained by restricting the connection length to $w = 1$. On one hand, this makes the experimental setup easier, since no all-to-all connections between the memories are required. On the other hand, from a theoretical point of view, the analysis of which memories are connected in which way, i.e., the underlying matching problem, becomes more complex when restricting the connection length to a finite regime (compare Sec. 3.4). In general, this result must be considered with caution, as the connection length within the N-partite quantum router is an artificially introduced concept. Since no assumptions are made here about a fixed arrangement of the quantum memories in an experimental quantum router setup, the analysis regarding the connection length refers to purely abstract considerations.

To optimally use the advantage of memory multiplexing, we analyze different storage and measurement strategies for the GHZ measurement. It turns out that the best strategy is to combine qubits with the highest fidelities first and remove qubits with lower fidelities when the fidelities fall below a certain cutoff. In future work, it would be interesting to extend the analysis to larger network structures with more than one central quantum router. The analytical results from our work are already general for arbitrary N and m. Note that the limitation is due to the complexity of the restricted underlying quantum router matching needed to simulate the quantum router. This shows again that finding good approximation algorithms for the weighted quantum router maximum matching for restricted connection length, as discussed in Sec. 3.4, is of great interest.

4.6 Publication

The main results from this chapter are published as a paper in *Physical Review A* in 2024 [KKB24]. The paper is attached to this work in Appendix B. Further work on this topic, as the improved analysis of the quantum router matching as well as the generalized equations for the QBERs and the fidelity of the generated GHZ state (Eq. (4.21), Eq. (4.22), and Eq. (4.19)) are part of the publication [BGK$^+$25] which is presented in Appendix A.

4.7 Personal contribution

The results based on the work [KKB24], as well as all the coding of all simulations, were done by me. Luis Gindorf came up with the generalized formulas for the QBERs and for the fidelities of the generated states with the $\left|GHZ_0^+\right\rangle$ by inferring them from concrete examples (i.e., for $N = 3$, $N = 4$, and $N = 5$ parties). The derivation of the general validity of these equations was done by me.

CHAPTER 5

LIMITATIONS OF THE ACHIEVABLE ROUTER RATE IN THE STATIONARY REGIME

5.1 Introduction

In the previous chapter, we showed that memory multiplexing helps increase the secret key rate. To get a fundamental understanding of the scalability of quantum networks with one central quantum router, we now focus on achievable router rates in the stationary regime. Motivated by the question of how many entangled states can be distributed per round in different network sizes, we develop analytical expressions to get limits on the router rate based on the number of parties and the number of available memories. With these expressions, we get an idea of how the router rate changes with an increasing number of participants and how a decrease in the router rate can be reduced by adding more memories per party.

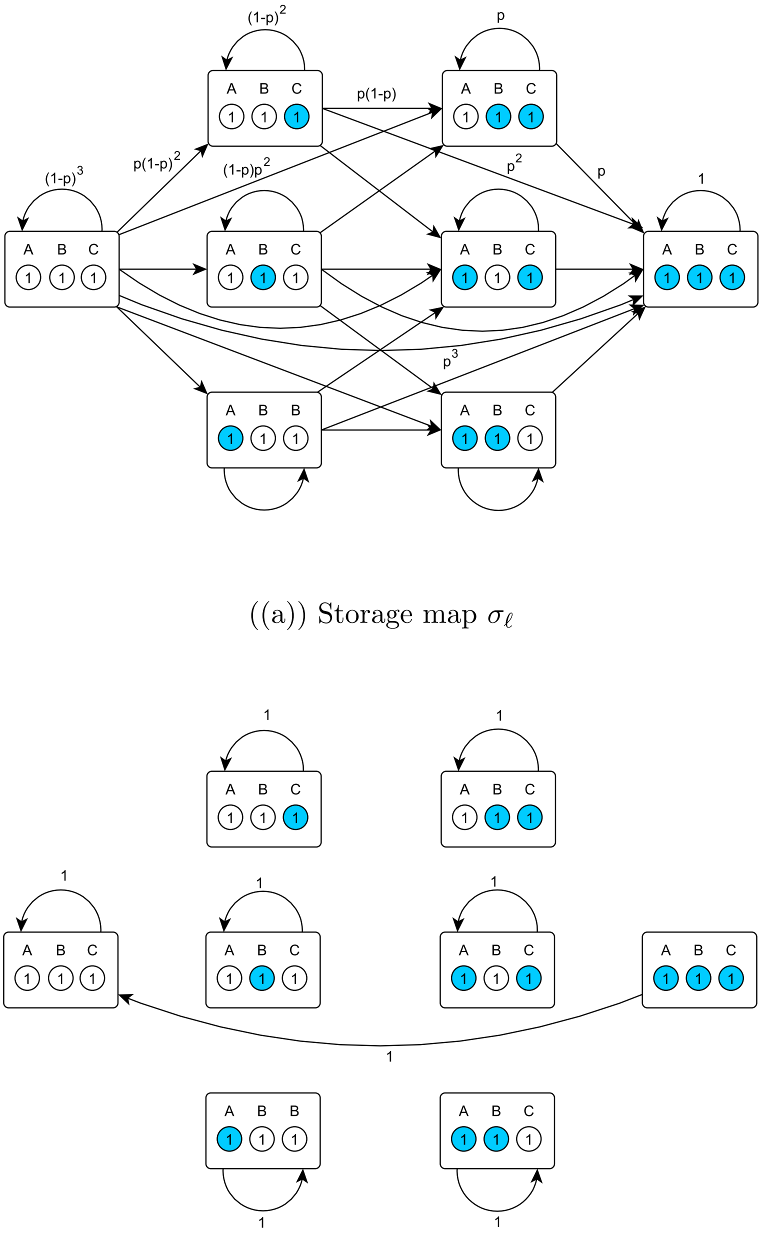
Previous works about quantum repeater chains analyzed the average waiting time a qubit has to wait until it is used for a Bell state measurement [SSvL19, KMSD19]. In contrast, we focus on the rate of entangled links distributed per round in the long-term regime in a multipartite quantum router. Vinay et al. analyzed the bipartite entanglement generation rate in repeater chains in [VK19]. Compared to [KSSvL23], we store qubits over the rounds and do not empty the memories after a successful GHZ measurement. With this, we analyze the advantage of not emptying the memories immediately.

The problem of finding the long-term router rate comes along with the task of finding the stationary distribution of a Markov chain. This can be done by solving a system of linear equations that describe the Markov process. Generally, this is numerically solvable for a given system of linear equations, i.e., for a fixed network size. Nevertheless, we prefer to have analytic solutions for arbitrary N and m to understand the dependency of the router rate on the network size. For the bipartite setup (i.e., $N = 2$), we get analytic results, while for the multipartite setup, we get very good approximations giving the router rate in the stationary limit.

5.2 More about Markov chains

Before making use of the theory of Markov chains, we repeat the main aspects on how we model the quantum router as a Markov process and discuss some main features. For a more detailed introduction, see Sec. 2.3.1. So far, we modeled the bipartite quantum repeater as a Markov chain by describing the configuration of the quantum repeater by a binary string $\mathcal{C} = \{0,1\}^{2m}$ of total length $2m$. For the N-partite quantum router, we make use of the same notation, now having a binary string $\mathcal{C} = \{0,1\}^{Nm}$ of total length Nm. Analogously, the state vector π is now a vector of length 2^{Nm}, where each entry π_i represents the probability of the quantum router to be in the i-th out of 2^{Nm} possible configurations. Note that a configuration denoted as $\mathcal{C}_i$ represents the binary string corresponding to the binary representation of i. The transition matrix T also increases in dimension to $2^{Nm} \times 2^{Nm}$.

For clarity, we give an example of the transition maps for a tripartite quantum router with one memory per party. The configurations that the quantum router can be in are as follows: $\mathcal{C}_0 = \{0,0,0\}$, $\mathcal{C}_1 = \{0,0,1\}$, $\mathcal{C}_2 = \{0,1,0\}$, $\mathcal{C}_3 = \{0,1,1\}$, $\mathcal{C}_4 = \{1,0,0\}$, $\mathcal{C}_5 = \{1,0,1\}$, $\mathcal{C}_6 = \{1,1,0\}$, and $\mathcal{C}_7 = \{1,1,1\}$. In Fig. 5.1, we show the transitions given by the storage map (see Fig. 5.1(a)) and the measurement map (see Fig. 5.1(b)) for the tripartite quantum router without multiplexing. For



((a)) Storage map σ_ℓ

((b)) Measurement map μ_ℓ

Figure 5.1: Transition maps for the tripartite quantum router with one memory per party. Along the arrows, the probabilities for the transitions are given. For a better overview, only one arrow of each possible transition between a state with $\tilde{m}_i$ initially filled memories to $\tilde{m}_f$ finally filled memories is given. All transitions from a configuration with the same $\tilde{m}_i$ and $\tilde{m}_f$ have the same probability, since the probability does not depend on the order of the memories. Arrows between two memory configurations that are not shown have zero probability. Memories shown in blue indicate that they are filled with a qubit.

this example, the matrix representing the storage map is given by:

$$\sigma_\ell = \begin{pmatrix} (1-p)^3 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ p(1-p)^2 & (1-p)^2 & 0 & 0 & 0 & 0 & 0 & 0 \\ p(1-p)^2 & 0 & (1-p)^2 & 0 & 0 & 0 & 0 & 0 \\ p^2(1-p) & p(1-p) & p^2(1-p) & 1-p & 0 & 0 & 0 & 0 \\ p(1-p)^2 & 0 & 0 & 0 & (1-p)^2 & 0 & 0 & 0 \\ p^2(1-p) & p(1-p) & 0 & 0 & p(1-p) & 1-p & 0 & 0 \\ p^2(1-p) & 0 & p(1-p) & 0 & p(1-p) & 0 & 1-p & 0 \\ p^3 & p^2 & p^2 & p & p^2 & p & p & 1 \end{pmatrix}. \tag{5.1}$$

Similar to the bipartite case, an entry of the measurement map is set to one if a GHZ measurement is performed (i.e., each party has at least one filled memory) and zero otherwise. The resulting matrix for the measurement map of the given example is as follows:

$$\mu_\ell = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{5.2}$$

Note that the storage map can always be written as a tensor product of the storage map for a single party. In the case of the given example where each party has only one memory available, the storage map for a single party is defined as $\sigma_\ell^{(1)} = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}$. The matrix from Eq. (5.1) is then given by $\sigma_\ell = \left(\sigma_\ell^{(1)}\right)^{\otimes 3}$ as $N = 3$ parties are considered here. On the contrary, it is not possible to generalize the measurement map. For each network size, one has to define how the memories are chosen for the GHZ measurement by following the underlying quantum router matching. Only for the case of no multiplexing (i.e., $m = 1$), it is possible to write the measurement map as

$$\mu_\ell = \mathbb{1}_{2^N} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes N} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes N} \tag{5.3}$$

since all configurations remain the same except the one where all memories are filled, i.e., configuration $\mathcal{C}_{2^N-1}$ changes to $\mathcal{C}_0$.

The transition matrix then follows from the concatenation of the storage map and the measurement map:

$$T = \mu_\ell \circ \sigma_\ell. \tag{5.4}$$

Given the transition matrix T and any initial configuration as the state vector π_{init}, we can calculate the probability distribution after s rounds as $\pi_s = T^s \pi_{init}$. In the asymptotic limit of infinitely many rounds (i.e., $s \to \infty$), we find the stationary distribution $\bar{\pi}$ as

$$\bar{\pi} = \lim_{s\to\infty} T^s \pi_{init}, \tag{5.5}$$

with $\bar{\pi} = T\bar{\pi}$. The stationary distribution $\bar{\pi}$ of the Markov chain representing the quantum router is unique as this Markov chain is *irreducible*. A Markov chain is irreducible if it has only one *recurrence class*. That means that any configuration of this recurrence class can reach any other configuration of the same recurrence class within a finite number of rounds. This is true for the quantum router setup, as any number of quantum memories can be filled in each round. Therefore, it is always possible to reach any configuration from the empty quantum router. Also, the empty quantum router can always be reached by filling all memories and performing GHZ measurements on all qubits later. By filling one memory per round, all combinations can be reached successively. As the total number of memories has to be finite, the number of rounds in which we can go from one configuration to another is also finite. Consequently, the stationary distribution is unique and, therefore, independent of the initial configuration given by the state vector π_{init}.

5.3 Results

In our work, we focus on the two setups of a multipartite quantum router: without multiplexing ($m = 1$) or with multiplexing ($m > 1$). Overall, we assume that the connection length is not restricted. Whenever there is at least one memory filled per party, a GHZ measurement is performed independently of the memory label.

5.3.1 No multiplexing

We first analyze the behavior of the router rate in the stationary regime of a quantum router without multiplexing. On an abstract level, this setup is similar to a bipartite quantum repeater chain. In a repeater chain of N segments, two end nodes are connected via $N-1$ quantum repeaters [BDCZ98]. Similar to the multipartite setup, one link exists between each station (i.e., no multiplexing in the repeater chains), and a link between the end nodes is only established if all intermediate links between the quantum repeaters exist. So far, the average waiting time [SSvL19, KMSD19], the entanglement generation rate (repeater rate) [VK19], and the exact rate for larger repeater chains [KSSvL23] have been analyzed. In our work, we show that the router rate, i.e., the entanglement generation rate for the multipartite quantum router without multiplexing, coincides with the results shown in [BPvL11].

We derive this result by considering the multipartite quantum router setup and not the repeater chain. Since the resulting Markov chain is irreducible and, therefore, leads to an unique stationary distribution $\bar{\pi} = \lim_{s\to\infty} T^s \pi_{init}$, we can calculate this stationary distribution by finding an expression for the term $\lim_{s\to\infty} T^s$. As the measurement map μ_ℓ can be generalized for the case without multiplexing (see Eq. (5.3)), a general expression for the transition matrix T for arbitrary N can also be found:

$$\begin{aligned} T' &= \sigma \circ \mu \\ &= \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}^{\otimes N} \left(\mathbb{1}_{2^N} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes N} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes N} \right) \qquad (5.6) \\ &\equiv \sigma + \sigma X_N, \qquad (5.7) \end{aligned}$$

with the storage map given by the tensor product of the storage map for a single memory $\sigma^{(1)} = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}$ and the term $X_N = \left(- \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes N} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes N} \right)$. Note that we here changed the order of the transition matrix from $T = \mu \circ \sigma$ to $T' = \sigma \circ \mu$, since this simplifies part of the calculation. As the stationary distribution does not depend on the initial distribution π_{init}, we can perform a measurement first (by applying μ) without changing the result.

To calculate the term T'^s, we make use of the following properties:

$$\begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \sigma \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} = \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \sigma^s \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} = 1 \tag{5.8}$$

and

$$\begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \sigma^s \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} = \left(1 - (1-p)^s\right)^N, \tag{5.9}$$

where $\begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}$ and $\begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} = \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}$. Thus, the term $X_N \sigma X_N$ can be simplified as follows:

$$\begin{aligned}
X_N \sigma X_N &= \left(- \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes N} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes N} \right) \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}^{\otimes N} \left(- \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes N} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes N} \right) \\
&= \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \\
&- \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \\
&- \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix} \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \\
&+ \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes N} \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \\
&= \left(\left(1 - (1-p)^s\right)^N - 1 \right) X_N \equiv \left(\wp_s - 1\right) X_N,
\end{aligned} \tag{5.10}$$

where we set $\wp_s = (1-(1-p)^s)^N$. Using this, the general term T'^s can be written as

$$T'^s = \sigma^s + \sum_{i=1}^{s} A_i X_N B_{s-i}, \tag{5.11}$$

with

$$B_{s-i} = \sigma^{s-i} \tag{5.12}$$

and

$$A_i = \sum_{j=1}^{i} c_j^{(i)} \sigma^j. \tag{5.13}$$

The prefactors $c_j^{(i)}$ are of the form:

$$c_j^{(i)} = \begin{cases} \sum_{l=1}^{i-1} c_1^{(l)} \left(\wp_{i-l} - 1\right), & j = 1 \\ c_{j-1}^{(i-1)}, & j > 1 \end{cases} \tag{5.14}$$

with the first prefactor being $c_1^{(1)} = 1$. We derive Eq. (5.11) from the concrete calculation of T'^2 and T'^3 and prove its general validity via induction.

For the stationary distribution in the asymptotic regime, we calculate $T'^{s\to\infty}\pi_{init}$ with the initial configuration chosen as $\pi_{init} = \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N}$, i.e., all memories are filled at the beginning. Since the goal is to determine the number of feasible GHZ measurements per round, it suffices to determine the probability that all memories are filled in the steady-state regime. This probability ($Prob[\ell = 1]$) is given by the last entry of the state vector $\bar{\pi}$. Therefore, we consider:

$$\begin{aligned}
Prob[\ell = 1] &= \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \bar{\pi} \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} T'^{s} \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N} \\
&= \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes N} \left[\sigma^{s} + \sum_{i=1}^{s} A_i X_N B_{s-i} \right] \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes N}
\end{aligned} \tag{5.15}$$

$$\begin{aligned}
&= 1 + \sum_{i=1}^{s} \sum_{j=1}^{i} c_j^{(i)} \left(\wp_j - 1 \right) \\
&= 1 + \sum_{i=2}^{s} \sum_{j=1}^{i-1} c_1^{(j)} (\wp_{i-j} - 1) = \sum_{j=1}^{s} c_1^{(j)}.
\end{aligned} \tag{5.16}$$

By setting $s \to \infty$ and rearranging the terms, we get the desired result:

$$Prob[\ell = 1] = \frac{1}{1 + \sum_{s=1}^{\infty} \left(1 - \left(1 - (1-p)^{s}\right)^{N} \right)} \equiv \langle \ell \rangle. \tag{5.17}$$

As this is the setup without multiplexing, the probability that one measurement is performed in a round equals the average number of GHZ measurements performed in each round, i.e., $\langle \ell \rangle = \sum_{\ell=0}^{1} \ell \cdot Prob[\ell] = Prob[\ell = 1]$.

This result is already known in the context of repeater chains [BPvL11]. As the abstract description of a quantum router without multiplexing and the bipartite repeater chain coincide, so do the results. Eq. (5.17) can be written as

$$\langle \ell \rangle = \left[\sum_{k=1}^{N} \frac{(-1)^{k+1}}{1-(1-p)^{k}} \binom{N}{k} \right]^{-1}, \tag{5.18}$$

which is known from probability theory: the term in brackets gives the expectation value of the maximum of N independent geometrically distributed random variables with success probability p [SR90]. In the context of the quantum router, this quantity gives the average waiting time to perform a GHZ measurement starting with an empty quantum router. The inverse of the average waiting time gives the desired value $\langle \ell \rangle$ since no multiplexing is considered here. Performing a GHZ measurement

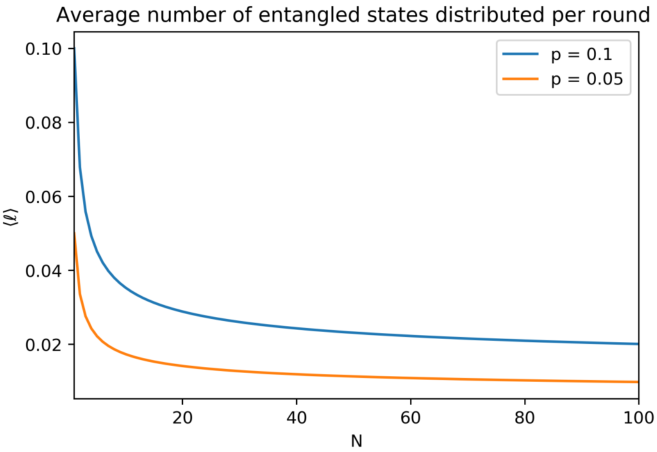


Figure 5.2: ([KTW+25]) Average number of GHZ measurements performed in the asymptotic limit for a quantum router without multiplexing for two different success probabilities $p = 0.1$ and $p = 0.05$. The calculation is done by following Eq. 5.17.

always returns the quantum router to the initial state of all memories being empty, and the quantum router is reset to the initial state.

The results show that the router rate decreases with an increasing number of communicating parties. An exemplary plot of the router rate as a function of the number of parties N for a fixed success probability $p = 0.1$ and $p = 0.05$ is shown in Fig. 5.2. Although the router rate decreases slowly with the number of parties (i.e., $\langle \ell \rangle \propto (\ln N)^-1$), the figure shows that for small N a significant drop in the router rate occurs. Increasing the number of communicating parties from $N = 2$ to $N = 3$ already leads to a decrease of about 32% in the router rate. This clearly demonstrates that adding quantum memories is a relevant step to counteract this drop in the router rate.

5.3.2 Multiplexing

Now, we consider the quantum router with multiplexing, i.e., $m > 1$. Generally, for a quantum router with N parties and m memories each, 2^{Nm} different memory configurations exist. To simplify the following discussion, and since we do not set any restrictions on the connection length, we can ignore the order of the memories of each party. This means that in the following, we will not consider the configuration itself but rather the number of filled memories $\tilde{m} = |m|$ per party. Then, we only consider the number of all configurations that have $\tilde{m}$ filled memories. This reduces the number of configurations to $(m+1)^N$. The state vector π is then a vector of length $(m+1)^N$, the storage map σ, the measurement map μ, and the transition matrix $T = \mu\sigma$ are then given by a $(m+1)^N \times (m+1)^N$ matrix. In the following, we denote a configuration as $\mathbf{k} = (\tilde{m}_A, \tilde{m}_{B_1}, \dots, \tilde{m}_{B_{N-1}})$ with $\tilde{m}_A$ for the number of filled memories on Alice's side and $\tilde{m}_{B_i}$ for all Bob's (B_i with $i \in \{1, \dots, N-1\}$) memories, respectively.

The difficulty of calculating the power of the transition matrix T is due to the fact that it is not possible to find a general expression for the storage and the measurement map. However, it is possible to write down the storage map for a fixed number of memories m and a single party denoted as $\sigma^{(1)}$. From this, the storage map for multiple parties N and fixed m can be calculated as the tensor product, i.e., $\sigma^{(N)} = \left(\sigma^{(1)}\right)^{\otimes N}$. The single-party matrix is given by:

$$\sigma^{(1)}_{\tilde{m}_i, \tilde{m}_f} = \begin{cases} 0, & \tilde{m}_i > \tilde{m}_f \\ \binom{m - \tilde{m}_i}{\tilde{m}_f - \tilde{m}_i} (1-p)^{m - \tilde{m}_f} p^{\tilde{m}_f - \tilde{m}_i}, & otherwise \end{cases} \tag{5.19}$$

with $\tilde{m}_i, \tilde{m}_f \in \{0, 1, \dots, m\}$ being the number of filled memories before ($\tilde{m}_i$) and after ($\tilde{m}_f$) storing all qubits that have arrived successfully. However, it is impossible to represent the measurement map via the tensor product for larger N. Here, for each choice of N and m, the matrix must be considered explicitly using the underlying matching. Consequently, it is only possible to generate the transition matrix independently for every specific network size. The system of linear equations of the form

$$\bar{\pi} = T\bar{\pi}, \tag{5.20}$$

with state vector $\bar{\pi}$ in the stationary regime and transition matrix $T = \mu\sigma$ can be solved for given N and m.

In our work, we provide an approximate equation that can be used to calculate the router rate for arbitrary N and m. With this, the dependency of the router rate on the network size (i.e., for any N and m) can be deduced.

Bipartite setup

In a first step, we focus on the bipartite scenario ($N = 2$) and derive an approximate expression for the router rate under the assumption that only small orders of success probability p are considered. This means that in every round, at most one qubit arrives at the quantum router and is stored successfully. Stored qubits that are not used for a GHZ measurement in a round are still kept in the memories for the next round. A valid transition in one round is then one of the following:

- with probability $2mp$, a first qubit is stored either on Alice's or Bob's side in the quantum router (here, we assume without loss of generality that Alice's memories are filled first),
- with probability $(m - \tilde{m})p$, another memory from party A is filled (assuming that $\tilde{m}$ memories are filled already),
- with probability mp, a memory is filled on Bob's side and, therefore, a GHZ measurement is performed (both parties then consume one qubit), or
- with remaining probability, nothing changes.

The stationary distribution can be maintained via the Markov chain tree theorem [LR86] that connects the probability of finding a specific configuration $\mathbf{k} \in \{(0,0),(1,0),\ldots,(m,0)\}$ (each represented by a node in the Markov chain) to the weights of its arborescences $A_{\mathbf{k}}$. That means, that for each node $\mathbf{k}$ in the graph representation of the Markov chain, we find a set of edges $A \subseteq E$ such that each other node $\mathbf{k}' \in V$ has a directed path to the chosen node $\mathbf{k}$ and all nodes except the chosen node $\mathbf{k}$ have exactly one outgoing edge. In this specific example of $N = 2$, each node $\mathbf{k}$ has exactly one arborescence $A_{\mathbf{k}}$. The weight of each arborescence, denoted by $\|A\|$, is defined as the product of the edge transition probabilities q_e for all $e \in A$, as described above:

$$\|A\| = \prod_{e \in A} q_e. \tag{5.21}$$

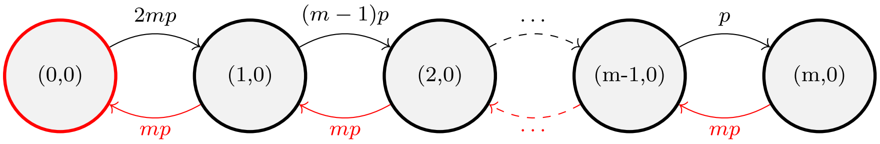


Figure 5.3: ([KTW+25]) Markov chain for the bipartite quantum router represented as a graph. The nodes give the different memory configurations $(\tilde{m}_A, \tilde{m}_B)$ giving the number of filled memories for Alice ($\tilde{m}_A$) and Bob ($\tilde{m}_B$), respectively. Associated with each edge is the transmission probability q to go from one configuration to another configuration. Since self-loops are not considered in the Markov chain tree theorem [LR86], these edges are not shown here. The set of edges colored in red represents the arborescence belonging to the node with configuration $(0, 0)$ (also colored in red).

The Markov chain representing the bipartite quantum repeater as well as the arborescence for node $(0, 0)$ are shown in Fig. 5.3. Due to the assumption of small p, exactly one GHZ measurement can be performed whenever there is at least one memory filled on Alice's side, and with probability mp, one memory is filled on Bob's side in the following round. To compute the probability of ending up in a configuration with at least one filled memory (i.e., $\mathbf{k} \in (1, 0), \ldots, (m, 0)$), we apply the Markov chain tree theorem:

$$\begin{aligned} Prob[\ell = 1] &= \langle \ell \rangle = 1 - \pi_{(0,0)} \\ &= 1 - \frac{\|A_{(0,0)}\|}{\sum_{k=0}^{m} \|A_{(k,0)}\|}. \end{aligned} \tag{5.22}$$

The router rate in the stationary regime then follows as:

$$\begin{aligned} R_\infty &= pm\frac{\langle \ell \rangle}{m} = p\langle \ell \rangle \\ &= p\left(1 - \frac{1}{1 + 2\sum_{k=1}^{m} m^{-k+1}\frac{(m-1)!}{(m-k)!}}\right). \end{aligned} \tag{5.23}$$

A comparison of the router rate obtained by this equation with the results from the simulation shows that the approach of small p is a lower bound, providing good results also for larger m.

Generalization to arbitrary N

A direct generalization of the Markov chain tree theorem to networks with more than two parties is impossible, even under the assumption of small p. The theorem is easily applicable for the bipartite setup since all roots $\mathbf{k}$ in the underlying graph have a single arborescence due to the simple graph structure obtained for $N = 2$ and small p. For the multipartite setup, the number of arborescences per root increases. Even in the simplest case of $N = 3$ and $m = 2$, each root $\mathbf{k}$ has three arborescences. This can be determined using Tutte's directed matrix-tree theorem [Tut01].

Nevertheless, it is possible to derive an approximate formula for the router rate for the general case. To do so, we make the following assumptions for a simplified model:

- The random number of performed GHZ measurements ℓ is substituted in the original model by a fixed number $\tilde{\ell}$ that is associated with the average value $\langle \ell \rangle$ in the stationary regime.
- The expectation value of the number of filled memories is allowed to be negative for each party. Only their stationary average values are required to be non-negative.
- The stationary occupation numbers giving the number of filled memories per party are given by the normal distribution.

The average number of GHZ measurements that are performed in the stationary regime results in:

$$\langle \ell \rangle = pm \left(\sqrt{\frac{\alpha^2 \beta}{4m} \ln N + 1} - \sqrt{\frac{\alpha^2 \beta}{4m} \ln N} \right)^2 \tag{5.24}$$

such that the router rate follows

$$\begin{aligned} R_\infty &= \frac{\langle \ell \rangle}{m} \\ &= p \left(\sqrt{\frac{\alpha^2 \beta}{4m} \ln N + 1} - \sqrt{\frac{\alpha^2 \beta}{4m} \ln N} \right)^2 , \end{aligned} \tag{5.25}$$

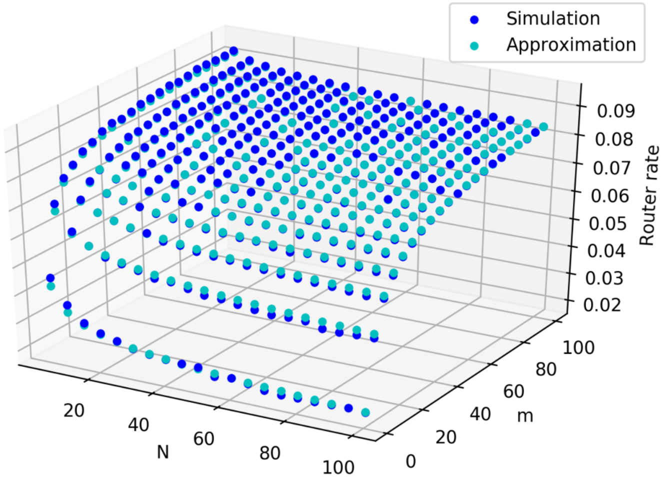


Figure 5.4: Comparison of the router rate between the approximation obtained here and the simulation of the quantum router. The plot covers different network sizes with varying N and m.

with $\beta = (1-p)/(2-p)$. For the parameter α we find

$$\alpha = \sqrt{2}\left(1 - \frac{\ln(4\pi \ln N) - 2\gamma}{4 \ln N}\right), \tag{5.26}$$

where $\gamma = 0.5572\ldots$ is the Euler-Mascheroni constant.

To verify our approach, we compare the router rate obtained with Eq. 5.25 with the simulations performed similarly as described in Sec. 4.3. Fig. 5.4 shows the router rate for different network sizes, either obtained via the approximation formula or the simulations. The plot shows that Eq. 5.25 leads to a good approximation of the router rate in the asymptotic regime. In Fig. 5.5, we show the scaling of the router rate with the network size for different networks up to $N = 150$ parties with $m = 100$ memories each. The router rate is calculated via the approximation formula. It turns out that the router rate initially increases linearly with the number of memories, while it saturates for larger network sizes. When each party has enough memories available, the router rate does not further increase. In Fig. 5.5, this threshold is shown with the red dots. The router rate reached at that configuration does not increase significantly (i.e., the difference becomes smaller than 0.0001) when

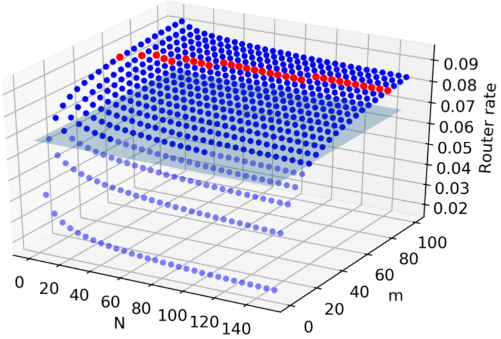


Figure 5.5: ([KTW$^+$25]) Scaling of the router rate calculated via Eq. (5.25) for larger networks. The line of red dots indicates the maximal achievable router rate for a given number of parties N. Adding more memories per party does not further increase the router rate, i.e., the rate increases less than linearly with the number of memories. The light blue plane shows the achievable router rate of the simplest case with $N = 2$ and $m = 1$. This threshold can be surpassed by increasing the number of memories per party for all network sizes considered here.

more memories are added. Additionally, we show that the router rate achieved in the simplest case, i.e., for $N = 2$ and $m = 1$ with $R_\infty = 0.068$, can be obtained in all network configurations considered here by adding more memories to each party. The threshold of that minimal achievable router rate in a network is shown by the light blue plane in Fig. 5.5.

In the case of large N, we find

$$\lim_{N\to\infty} R_\infty \approx \frac{pm}{\alpha^2\beta \ln N} \propto (\ln N)^{-1} \,. \tag{5.27}$$

This goes along with the results we get for the case of no multiplexing in Sec. 5.3.1. For large m, we see that the router rate converges, i.e., it follows

$$\lim_{m\to\infty} R_\infty \approx p. \tag{5.28}$$

The effect of adding more memories does not change the router rate, as we have already seen from the simulations.

5.4 Conclusion

With this work, we derive the relation between the router rate and the network size of a star graph with N parties and m memories each. Additionally, the results can be used to derive the relation to the success probability p. For large m, the router rate converges to p, i.e., the router rate mainly depends on the success probability for large m. Here, mp forms an upper bound on the achievable router rate for general network sizes with arbitrary N and m. When considering only parallel connections (i.e., $w = 0$), the router rate is given as $R_\infty = m\langle \ell_1 \rangle$ with router rate $\langle \ell_1 \rangle$ for the case without multiplexing. Then, the rate depends only on the achievable rate for a single memory $\langle \ell_1 \rangle$ and the number of parallel connections m that can, in principle, be created. Note that $m\langle \ell_1 \rangle$ gives a lower bound on the router rate for each N.

We find that the multipartite quantum router without multiplexing (i.e., $m = 1$) coincides with the description of a bipartite repeater chain with $N - 1$ quantum repeaters. The results show that even though the router rate decreases slowly with the number of communicating parties N ($R_\infty \propto \ln(N)^{-1}$), the drop that goes along with small N is significant. This result clearly demonstrates the importance of quantum memories to counteract the rate reduction with increasing N. Secondly, we derive an approximate formula for the quantum router with multiplexing ($m > 1$). This shows the dependency of the router rate on the network size (N, m) and the success probability p of creating bipartite Bell pairs. The results show that the router rate saturates for large m, which means that the average number of GHZ state measurements performed does not grow faster than linearly with the number of memories m.

The results of this work can be used to plan quantum networks and estimate achievable router rates depending on the network size. In particular, the optimal number of quantum memories can be determined, especially when the number of participating parties increases, thereby decreasing the router rate. Due to the convergence of the router rate that arises for large m, it would be interesting to investigate alternative protocols that use additional quantum memories. The integration of entanglement distillation of the bipartite Bell pairs [DEJ$^+$96, DBCZ99, BAKB13] could be one application that could make use of additional quantum memories. This research direction is left for future work.

5.5 Publication

The results of this chapter are published in our work in [KTW$^+$25]. The arXiv version of this paper is attached to this work in Appendix C.

5.6 Personal contribution

The derivation of the router rate for a multipartite quantum router without multiplexing was done in equal parts by Nikolai Wyderka and me. In doing so, Nikolai Wyderka essentially inferred the Eqs (5.11) to (5.14). The proofs of their general validity were done by me. Anton Trushechkin performed the transformation to the known results given in Eq. (5.18). The results for the bipartite quantum repeater were obtained by Anton Trushechkin and me in equal parts. The generalization to networks with arbitrary N and m was done by Anton Trushechkin. The comparison of the approximation formula with the simulations was executed by me. The simulations that were used for the validation of the approximation formulas were developed by me in the context of the previous work (see Sec. 4.3). The publication was written by Anton Trushechkin and me in equal parts with all calculations and plots being performed by me.

CHAPTER 6

DISCUSSION

Entanglement is one of the fundamental resources needed in quantum networks for communication and secret key distribution. To distribute entangled states over larger distances, intermediate stations called quantum repeaters are used. By performing Bell state measurements, these entangled links can be distributed among the parties.

In our work, we consider a generalization to the multipartite quantum router. Similar to [CJKK07], we introduce memories for multiplexing. We investigate the effect of the quantum memories on the qubit fidelities and optimize the secret key rate by exploring different matching strategies similar to the bipartite quantum repeater in [AKB14a]. We show that the secret key rate is optimal when combining the freshest qubits first and removing older qubits with a fidelity smaller than a certain threshold – defined as the cutoff.

To find the best matching strategy, we have properly defined and analyzed the underlying graph-theoretical problem of the quantum router matching. We show that the corresponding decision problem is generally $\mathcal{NP}$-complete, from which hardness for the maximum quantum router matching follows. However, for some special cases, we find algorithms that solve the problem in polynomial time. Furthermore, we develop an approximation algorithm for the general unweighted maximum quantum router matching. This solves the matching problem but does not always guarantee maximum cardinality.

In addition to the influence of the quantum memories on the qubit fidelity, we also examine the dependence of the entanglement generation rate (or router rate) on the number of quantum memories per party. We show that by increasing the number of quantum memories per party, the decrease in the router rate, which comes along with an increasing number of parties, can be counteracted. To this end, we derive a general expression for the router rate in the asymptotic limit as a function of the number of parties and memories. This expression allows us to calculate in advance the maximum number of entangled states that can be distributed per round in the asymptotic limit.

Our work can be used to plan quantum router networks with a central quantum router for entanglement distribution or conference key agreement. The matching algorithms we propose can be integrated to determine optimal qubit combinations to improve both the number of distributed entangled states per round and the secret key rate in quantum key distribution. Moreover, our analysis of the router rate in dependence on the network size (Eq. (5.25)) offers valuable guidance for planning and scaling quantum networks.

This work reveals further research directions, and interesting questions remain. It is worthwhile to investigate which rates result in larger, more complex networks that incorporate multiple quantum routers. Identifying effective strategies for optimizing the secret key rate in such extended architectures is also of great interest. To simulate large-scale networks efficiently, it is advisable to study the weighted matching problem further and develop a suitable approximation algorithm for it.

Additionally, new strategies for improving the secret key rate can be explored. As discussed in [BAKB13], purification protocols offer one such approach. Established bipartite purification methods [DBCZ99, DEJ$^+$96] could be applied prior to entanglement swapping at the router. For example, all available quantum memories could be used for purification, or multiple purification processes could be performed in parallel, allowing the resulting parallel links to still support multiplexing. Alternatively, novel multipartite purification strategies could be developed and integrated into the protocol after performing the entanglement swapping.

Overall, our work outlines optimization strategies for both the router and secret key rates in quantum networks, using a central quantum router for entanglement distribution and conference key agreement. These results provide a foundation for further investigations into the performance of larger, more complex network topologies. Additionally, they can be extended by incorporating alternative noise models beyond depolarization, as well as by integrating advanced techniques such as entanglement purification.

BIBLIOGRAPHY

[ABDR04] A. Ambainis, H. Buhrman, Y. Dodis, and H. Rohrig. Multiparty quantum coin flipping. In *Proceedings. 19th IEEE Annual Conference on Computational Complexity, 2004.*, pages 250–259. IEEE, 2004.

[Abr14] S. Abruzzo. *Long distance quantum key distribution with quantum repeaters*. PhD thesis, Heinrich-Heine-Universität Düsseldorf, 2014.

[AKB14a] S. Abruzzo, H. Kampermann, and D. Bruß. Finite-range multiplexing enhances quantum key distribution via quantum repeaters. *Physical Review A*, 89(1):012303, 2014.

[AKB14b] S. Abruzzo, H. Kampermann, and D. Bruß. Measurement-device-independent quantum key distribution with quantum memories. *Physical Review A*, 89(1):012301, 2014.

[AKS02] M. Agrawal, N. Kayal, and N. Saxena. Primes is in P. *Annals of Mathematics*, 160:781–793, 2002.

[AMO93] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows: Theory, Algorithms, and Applications*. JSTOR, 1993.

[ARW23] G. Avis, F. Rozpedek, and S. Wehner. Analysis of multipartite entanglement distribution using a central quantum-network node. *Physical Review A*, 107(1):012609, 2023.

[BAKB13] S. Bratzik, S. Abruzzo, H. Kampermann, and D. Bruß. Quantum repeaters and quantum key distribution: The impact of entanglement distillation on the secret key rate. *Physical Review A*, 87(6):062335, 2013.

[BB84] C. H. Bennett and G. Brassard. Quantum cryptography: Public key distribution and coin tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*, pages 175–179. IEEE Press, 1984.

[BBM92] C. H. Bennett, G. Brassard, and N. D. Mermin. Quantum cryptography without Bell's theorem. *Physical Review Letters*, 68(5):557–559, 1992.

[BDCZ98] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller. Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication. *Physical Review Letters*, 81(26):5932–5935, 1998.

[Bel64] J. S. Bell. On the einstein podolsky rosen paradox. *Physics Physique Fizika*, 1(3):195–200, Nov 1964.

[BGK$^+$25] D. Bruß, L. Gindorf, J. A. Kunzelmann, C. Laußmann, and J. Rothe. On Matching in Multipartite Quantum Routers. In *Proceedings of (QUASAR '25)*. ACM, 2025. Accepted for presentation, proceedings forthcoming.

[Blo29] F. Bloch. Über die Quantenmechanik der Elektronen in Kristallgittern. *Zeitschrift fur Physik*, 52(7-8):555–600, 1929.

[Boh13] N. Bohr. On the constitution of atoms and molecules. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 26(151):1–25, 1913.

[BPvL11] N. K. Bernardes, L. Praxmeyer, and P. van Loock. Rate analysis for a hybrid quantum repeater. *Physical Review A*, 83(1):012323, 2011.

[BRM$^+$20] M. K. Bhaskar, R. Riedinger, B. Machielse, D. S. Levonian, C. T. Nguyen, E. N. Knall, H. Park, D. Englund, M. Lončar, D. D. Sukachev, and M. D. Lukin. Experimental demonstration of memory-enhanced quantum communication. *Nature*, 580(7801):60–64, 2020.

[Bru25] D. Bruß. Editorial: Celebrating the first century of quantum physics and preparing for the next one. *Physical Review Letters*, 134:150001, 2025.

[BVK98] S. Bose, V. Vedral, and P. L. Knight. Multiparticle generalization of entanglement swapping. *Physical Review A*, 57(2):822–829, 1998.

[CDRF$^+$05] C.-W. Chou, H. De Riedmatten, D. Felinto, S. V. Polyakov, S. J. Van Enk, and H. J. Kimble. Measurement-induced entanglement for excitation stored in remote atomic ensembles. *Nature*, 438(7069):828–832, 2005.

[CJKK07] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy. Multiplexed memory-insensitive quantum repeaters. *Physical Review Letters*, 98(6):060502, 2007.

[CKD$^+$21] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpedek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres-Knoop, D. Elkouss, and S. Wehner. Netsquid, a network simulator for quantum information using discrete events. *Communications Physics*, 4(1):164, 2021.

[CL08] K. Chen and H.-K. Lo. Multi-partite quantum cryptographic protocols with noisy GHZ states. *arXiv preprint quant-ph/0404133*, 2008.

[CMJ$^+$05] T. Chanelière, D. N. Matsukevich, S. D. Jenkins, S.-Y. Lan, T. A. B. Kennedy, and A. Kuzmich. Storage and retrieval of single photons transmitted between remote quantum memories. *Nature*, 438(7069):833–836, 2005.

[Coo71] S. A. Cook. The complexity of theorem-proving procedures. In *Proceedings of the Third Annual ACM Symposium on Theory of Computing*, STOC '71, page 151–158. Association for Computing Machinery, 1971.

[Cyg13] M. Cygan. Improved approximation for 3-dimensional matching via bounded pathwidth local search. In *2013 IEEE 54th Annual Symposium on Foundations of Computer Science*, pages 509–518. IEEE, 2013.

[DBCZ99] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller. Quantum repeaters based on entanglement purification. *Physical Review A*, 59(1):169–181, 1999.

[DEJ$^+$96] D. Deutsch, A. Ekert, R. Jozsa, C. Macchiavello, S. Popescu, and A. Sanpera. Quantum privacy amplification and the security of quantum cryptography over noisy channels. *Physical Review Letters*, 77(13):2818–2821, 1996.

[DH76] W. Diffie and M. E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, 22(6):644–654, 1976.

[DHR02] M. J. Donald, M. Horodecki, and O. Rudolph. The uniqueness theorem for entanglement measures. *Journal of Mathematical Physics*, 43(9):4252–4272, 2002.

[DLCZ01] L.-M. Duan, M. D. Lukin, J. I. Cirac, and P. Zoller. Long-distance quantum communication with atomic ensembles and linear optics. *Nature*, 414(6862):413–418, 2001.

[Ehr11] P. Ehrenfest. Welche Züge der Lichtquantenhypothese spielen in der Theorie der Wärmestrahlung eine wesentliche Rolle? *Annalen der Physik*, 341(11):91–118, 1911.

[EPR35] A. Einstein, B. Podolsky, and N. Rosen. Can quantum-mechanical description of physical reality be considered complete? *Physical review*, 47(10):777–780, 1935.

[FC72] S. J. Freedman and J. F. Clauser. Experimental test of local hidden-variable theories. *Physical Review Letters*, 28(14):938–941, 1972.

[Fed24] Federal Office for Information Security (BSI). Technical Guideline BSI-TR-02102-1: Cryptographic mechanisms: Recommendations and key lengths, 2024. Accessed: 2025-04-10.

[FVMH19] N. Friis, G. Vitagliano, M. Malik, and M. Huber. Entanglement certification from theory to experiment. *Nature Reviews Physics*, 1(1):72–87, 2019.

[GHZ89] D. M. Greenberger, M. A. Horne, and A. Zeilinger. Going beyond Bell's theorem. In *Bell's theorem, quantum theory and conceptions of the universe*, pages 69–72. Springer, 1989.

[GKB18] F. Grasselli, H. Kampermann, and D. Bruß. Finite-key effects in multipartite quantum key distribution protocols. *New Journal of Physics*, 20(11):113014, 2018.

[GMM$^+$24] F. Gu, S. G. Menon, D. Maier, A. Das, T. Chakraborty, W. Tittel, H. Bernien, and J. Borregaard. Hybrid quantum repeaters with ensemble-based quantum memories and single-spin photon transducers. *arXiv preprint arXiv:2401.12395*, 2024.

[Gra21] F. Grasselli. *Quantum Cryptography. From Key Distribution to Conference Key Agreement.* Quantum Science and Technology. Springer, 2021.

[GS22] W. Gerlach and O. Stern. Der experimentelle Nachweis der Richtungsquantelung im Magnetfeld. *Zeitschrift für Physik*, 9(1):349–352, 1922.

[GT09] O. Gühne and G. Tóth. Entanglement detection. *Physics Reports*, 474(1–6):1–75, 2009.

[Har82] J. Hartmanis. Computers and Intractability: A Guide to the Theory of NP-Completeness (Michael R. Garey and David S. Johnson). *SIAM Review*, 24(1):90–91, 1982.

[HBB99] M. Hillery, V. Bužek, and A. Berthiaume. Quantum secret sharing. *Physical Review A*, 59(3):1829–1834, 1999.

[HHHH09] R. Horodecki, P. Horodecki, M. Horodecki, and K. Horodecki. Quantum entanglement. *Reviews of Modern Physics*, 81(2):865–942, 2009.

[HK71] J. E. Hopcroft and R. M. Karp. An n5/2 Algorithm for Maximum Matchings in Bipartite Graphs. In *12th Annual Symposium on Switching and Automata Theory (SWAT 1971)*, pages 122–125. IEEE, 1971.

[HRP$^+$06] P. A. Hiskett, D. Rosenberg, C. G. Peterson, R. J. Hughes, S. Nam, A. E. Lita, A. J. Miller, and J. E. Nordholt. Long-distance quantum key distribution in optical fibre. *New Journal of Physics*, 8(9):193, 2006.

[JAK25] L. Gindorf D. Bruß J. Rothe J. A. Kunzelmann, C. Laußmann. Opening the Black Box of Quantum Router Matching. Manuscript in preparation, 2025.

[JFZ22] Z. Ji, P. Fan, and H. Zhang. Entanglement swapping for Bell states and Greenberger–Horne–Zeilinger states in qubit systems. *Physica A: Statistical Mechanics and its Applications*, 585:126400, 2022.

[JTN$^+$09] L. Jiang, J. M. Taylor, K. Nemoto, W. J. Munro, R. Van Meter, and M. D. Lukin. Quantum repeater with encoding. *Physical Review A*, 79(3):032325, 2009.

[Kar72] R. M. Karp. *Reducibility among Combinatorial Problems*, pages 85–103. Springer US, 1972.

[KCM$^+$23] V. Krutyanskiy, M. Canteri, M. Meraner, J. Bate, V. Krcmarsky, J. Schupp, N. Sangouard, and B. P. Lanyon. Telecom-wavelength quantum repeater node based on a trapped-ion processor. *Physical Review Letters*, 130(21):213601, 2023.

[KGR05] B. Kraus, N. Gisin, and R. Renner. Lower and upper bounds on the secret-key rate for quantum key distribution protocols using one-way classical communication. *Physical Review Letters*, 95(8):080501, 2005.

[KKB24] J. A. Kunzelmann, H. Kampermann, and D. Bruß. Multipartite multiplexing strategies for quantum routers. *Physical Review A*, 110(3):032617, 2024.

[KMSD19] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling. Practical figures of merit and thresholds for entanglement distribution in quantum networks. *Physical Review Research*, 1(2):023032, 2019.

[KSSvL23] L. Kamin, E. Shchukin, F. Schmidt, and P. van Loock. Exact rate analysis for quantum repeaters with imperfect memories and entanglement swapping as soon as possible. *Physical Review Research*, 5(2):023086, May 2023.

[KT06] J. Kleinberg and É. Tardos. *Algorithm Design.* Pearson Education India, 2006.

[KTW$^+$25] J. A. Kunzelmann, A. Trushechkin, N. Wyderka, H. Kampermann, and D. Bruß. Multiplexed multipartite quantum repeater rates in the stationary regime. *arXiv preprint arXiv:2505.18031*, 2025.

[LCA05] H.-K. Lo, H. F. Chau, and M. Ardehali. Efficient quantum key distribution scheme and a proof of its unconditional security. *Journal of Cryptology*, 18:133–165, 2005.

[Lev73] L. Levin. Universal sorting problems. *Problemy Peredaci Informacii*, 9:115–116, 1973. In Russian. English translation in *Problems of Information Transmission*, 9:265–266, 1973.

[LFL$^+$23a] C.-L. Li, Y. Fu, W.-B. Liu, Y.-M. Xie, B.-H. Li, M.-G. Zhou, H.-L. Yin, and Z.-B. Chen. All-photonic quantum repeater for multipartite entanglement generation. *Optics Letters*, 48(5):1244–1247, 2023.

[LFL$^+$23b] C.-L. Li, Y. Fu, W.-B. Liu, Y.-M. Xie, B.-H. Li, M.-G. Zhou, H.-L. Yin, and Z.-B. Chen. Breaking universal limitations on quantum conference key agreement without quantum memory. *Communications Physics*, 6(1):122, 2023.

[LP09] L. Lovász and M. D. Plummer. *Matching theory*, volume 367. American Mathematical Soc., 2009.

[LR86] F. Leighton and R. Rivest. Estimating a probability using finite memory. *IEEE Transactions on Information Theory*, 32(6):733–742, 1986.

[LYDS18] M. Lucamarini, Z. Yuan, J. F. Dynes, and A. J. Shields. Overcoming the rate–distance limit of quantum key distribution without quantum repeaters. *Nature*, 557(7705):400–403, 2018.

[LZJ$^+$23] Y. Liu, W.-J. Zhang, C. Jiang, J.-P. Chen, C. Zhang, W.-X. Pan, D. Ma, H. Dong, J.-M. Xiong, C.-J. Zhang, H. Li, R.-C. Wang, J. Wu, Y. Chen T, L. You, X.-B. Wang, Q. Zhang, and J.-W. Pan. Experimental twin-field quantum key distribution over 1000 km fiber distance. *Physical Review Letters*, 130(21):210801, 2023.

[LZW$^+$21] C. Li, S. Zhang, Y. Wu, N. Jiang, Y. Pu, and L. M. Duan. Multi-cell atomic quantum memory as a hardware-efficient quantum repeater node. *PRX Quantum*, 2(4):040307, 2021.

[MEW23] J. Memmen, J. Eisert, and N. Walk. Advantage of multi-partite entanglement for quantum cryptography over long and short ranged networks. *arXiv preprint arXiv:2312.13376*, 2023.

[MGKB20] G. Murta, F. Grasselli, H. Kampermann, and D. Bruß. Quantum conference key agreement: A review. *Advanced Quantum Technologies*, 3(11):2000025, 2020.

[MLK$^+$16] S. Muralidharan, L. Li, J. Kim, N. Lütkenhaus, M. D. Lukin, and L. Jiang. Optimal architectures for long distance quantum communication. *Scientific reports*, 6(1):20463, 2016.

[MT09] S. Meyn and R. L. Tweedie. *Markov Chains and Stochastic Stability*. Cambridge Mathematical Library. Cambridge University Press, 2nd edition, 2009.

[NC10] M. A. Nielsen and I. L. Chuang. *Quantum Computation and Quantum Information*. Cambridge University Press, 2010.

[Net24a] NetworkX Developers. *NetworkX Documentation: maximal_matching*, 2024. Accessed: 2025-04-03.

[Net24b] NetworkX Developers. networkx.algorithms.flow.maximum_flow — networkx documentation, 2024. Accessed: 2025-04-17.

[NVGT20] P. Nain, G. Vardoyan, S. Guha, and D. Towsley. On the analysis of a multipartite entanglement distribution switch. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 4(2):1–39, 2020.

[NVGT22] P. Nain, G. Vardoyan, S. Guha, and D. Towsley. Analysis of a tripartite entanglement distribution switch. *Queueing Systems*, 101(3):291–328, 2022.

[PJC$^+$17] Y. Pu, N. Jiang, W. Chang, H. Yang, C.-F. Li, and L. M. Duan. Experimental realization of a multiplexed quantum memory with 225 individually accessible memory cells. *Nature Communications*, 8(1):15359, 2017.

[PLOB17] S. Pirandola, R. Laurenza, C. Ottaviani, and L. Banchi. Fundamental limits of repeaterless quantum communications. *Nature Communications*, 8(1):15043, 2017.

[Pyt24] Python Software Foundation. itertools — functions creating iterators for efficient looping, 2024. Accessed: 2025-04-15.

[Rot05] J. Rothe. *Complexity Theory and Cryptology: An Introduction to Cryptocomplexity*. Springer, 2005.

[RSA78] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[Rud52] H. D. Ruderman. Two new inequalities. *The American Mathematical Monthly*, 59(1):29–32, 1952.

[SBPC$^+$09] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev. The security of practical quantum key distribution. *Reviews of Modern Physics*, 81(3):1301–1350, 2009.

[Sch35] E. Schrödinger. Die gegenwärtige Situation in der Quantenmechanik. *Naturwissenschaften*, 23(50):844–849, 1935.

[SdRA$^+$07] C. Simon, H. de Riedmatten, M. Afzelius, N. Sangouard, H. Zbinden, and N. Gisin. Quantum repeaters with photon pair sources and multimode memories. *Physical Review Letters*, 98(19):190503, 2007.

[Sha49] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28(4):656–715, 1949.

[Sho94] P.W. Shor. Algorithms for quantum computation: discrete logarithms and factoring. In *Proceedings 35th Annual Symposium on Foundations of Computer Science*, pages 124–134. IEEE, 1994.

[Sho97] P. W. Shor. Polynomial-time algorithms for prime factorization and discrete logarithms on a quantum computer. *SIAM Journal on Computing*, 26(5):1484–1509, 1997.

[SP00] P. W. Shor and J. Preskill. Simple proof of security of the BB84 quantum key distribution protocol. *Physical Review Letters*, 85(2):441–444, 2000.

[SR90] W. Szpankowski and V. Rego. Yet another application of a binomial recurrence order statistics. *Computing*, 43(4):401–410, 1990.

[SSdRG11] N. Sangouard, C. Simon, H. de Riedmatten, and N. Gisin. Quantum repeaters based on atomic ensembles and linear optics. *Reviews of Modern Physics*, 83(1):33–80, 2011.

[SSvL19] E. Shchukin, F. Schmidt, and P. van Loock. Waiting time in quantum repeaters with probabilistic entanglement swapping. *Physical Review A*, 100(3):032322, 2019.

[SWV$^+$09] D. Stucki, N. Walenta, F. Vannel, R. T. Thew, N. Gisin, H. Zbinden, S. Gray, C. R. Towery, and S. Ten. High rate, long-distance quantum key distribution over 250 km of ultra low loss fibres. *New Journal of Physics*, 11(7):075003, 2009.

[Ten23] L. A. Tendick. *Interplay of Quantum Resources in Bell-type Scenarios.* PhD thesis, Heinrich-Heine-Universität Düsseldorf, 2023.

[Tur37] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, s2-42(1):230–265, 1937.

[Tur96] A. M. Turing. Intelligent machinery, a heretical theory. *Philosophia Mathematica*, 4(3):256–260, 1996.

[Tut01] W. T. Tutte. *Graph theory*, volume 21. Cambridge University Press, 2001.

[VGNT21a] G. Vardoyan, S. Guha, P. Nain, and D. Towsley. On the capacity region of bipartite and tripartite entanglement switching. *SIGMETRICS Perform. Eval. Rev.*, 48(3):45–50, 2021.

[VGNT21b] G. Vardoyan, S. Guha, P. Nain, and D. Towsley. On the stochastic analysis of a quantum entanglement distribution switch. *IEEE Transactions on Quantum Engineering*, 2:1–16, 2021.

[VK19] S. E. Vinay and P. Kok. Statistical analysis of quantum-entangled-network generation. *Physical Review A*, 99(4):042313, 2019.

[VM14] R. Van Meter. *Quantum networking.* John Wiley & Sons, 2014.

[VPRK97] V. Vedral, M. B. Plenio, M. A. Rippin, and P. L. Knight. Quantifying entanglement. *Physical Review Letters*, 78(12):2275–2279, 1997.

[WHW$^+$24] J. Wallnöfer, F. Hahn, F. Wiesner, N. Walk, and J. Eisert. Faithfully simulating near-term quantum repeaters. *PRX Quantum*, 5(1):010351, 2024.

[Wil19] D. P. Williamson. *Network flow algorithms.* Cambridge University Press, 2019.

[Wol21] R. Wolf. Quantum key distribution. *Lecture notes in physics*, 988, 2021.

[WUZ$^+$17] Y. Wang, M. Um, J. Zhang, S. An, M. Lyu, J.-N. Zhang, L. M. Duan, D. Yum, and K. Kim. Single-qubit quantum memory exceeding ten-minute coherence time. *Nature Photonics*, 11(10):646–650, 2017.

[WZ82] W. K. Wootters and W. H. Zurek. A single quantum cannot be cloned. *Nature*, 299(5886):802–803, 1982.

[ZLJY23] L. Zhou, J. Lin, Y. Jing, and Z. Yuan. Twin-field quantum key distribution without optical frequency dissemination. *Nature Communications*, 14(1):928, 2023.

LIST OF FIGURES

LIST OF SYMBOLS

Quantum systems

$\mathcal{H}$	Hilbert space
$\|\psi\rangle$	Pure state as a ket-vector in the Hilbert space
$\mathcal{H}^*$	Dual of the Hilbert space
$\langle\psi\|$	Pure state as a bra-vector in the dual space
$\mathbb{C}$	Complex field
$(\cdot)^\dagger$	Hermitian conjugate
α, β	Complex numbers with complex conjugates $\bar{\alpha}, \bar{\beta}$
$\langle\phi\|\psi\rangle$	Inner product between state vectors $\|\phi\rangle, \|\psi\rangle$
ρ	Density operator or density matrix
d	Dimension of the the quantum system
γ, θ, φ	Real numbers of the Bloch sphere representation of a qubit
n	Number of quantum systems (Hilbert spaces) in a composed quantum system

$\otimes$	Tensor product
tr_i	Partial trace over system i
U	Unitary operator
M_m	Measurement operator with measurement outcome m
$P(m)$	Probability of measurement outcome m
M	Observable with eigenvalue m
P_m	Projector

Entanglement

$\|\phi^{\pm}\rangle, \|\psi^{\pm}\rangle$	Bell states
F	Fidelity
X, Y, Z	Pauli matrices
H	Hadamard gate
U_{CN}	Controlled not (CNOT) gate
A_1, A_2	Two qubits of a Bell pair of party A
B_1, B_2	Two qubits of a Bell pair of party B
a_i, b_i	Measurement outcomes of Bell measurements of qubit i from Party A, B
n	Number of qubits in the multipartite entangled quantum system
N	Number of parties in the multipartite setup
A	Center party with special role in the GHZ state measurement
B_i	Peer nodes with $i \in \{1, 2, \ldots, N-1\}$

Quantum networks

N	Number of parties in a network
α	Attenuation coefficient, here chosen to be $\alpha = 0.2$ dB/km
p	Probability to successfully generate, distribute, and store a Bell pair between two parties
d	Distance between two parties, i.e., length of quantum channel

Quantum repeater/router with memory multiplexing

A	Party A: the central party in the quantum router, due to a special role it has in the GHZ measurement procedure performed. In the bipartite setup, A has no special role
B_i	Parties B_i with $i \in \{1, \ldots, N-1\}$ in the N-partite quantum router. Also called the peers. For the bipartite quantum repeater, there is only one peer, i.e., Bob (B)
m	Number of memories per party
$\tilde{m}_\iota$	Number of filled memories per party $\iota \in \{A, B_1, \ldots, B_{N-1}\}$
w	Connection length with $w = m - 1$: full-range multiplexing, $1 \leq w < m - 1$: finite-range multiplexing, and $w = 0$: no multiplexing
ℓ	Number of Bell state measurements (GHZ measurements) performed in the quantum repeater (quantum router) for a given memory configuration
$\langle \ell \rangle$	Average number of performed Bell state measurements
$\langle \ell_1 \rangle$	Router rate of the setup without multiplexing, i.e., $m = 1$
$\tilde{\ell}$	Fixed number of performed GHZ state measurements used in the simplified model

s	Protocol round with current round s_c
R	Repeater/router rate
$\mathcal{C}$	Memory configuration of the quantum repeater, given as a bit string
c	Single bit in bit string $\mathcal{C}$ representing one quantum memory
π	State vector with entries π_i giving the probability for a specific configuration $\mathcal{C}_i$
$\bar{\pi}$	Stationary distribution of the state vector π in the asymptotic regime
T	Transition matrix given as $T = \mu\sigma$
T'	Transition matrix given as $T' = \sigma\mu$
σ_ℓ	Storage map
$\sigma^{(1)}$	Storage map for a single party
μ_ℓ	Measurement map
$\mathcal{H}_w^m(\ell)$	Set of configurations that leads to ℓ Bell state measurements
p_{BSM}	Success probability for performing an Bell state measurement
$\mathbf{k}$	Configuration of the quantum router given by the number of filled memories per party, i.e., $\mathbf{k} = (\tilde{m}_A, \tilde{m}_{B_1}, \dots, \tilde{m}_{B_{N-1}})$. Each such configuration represents one node in the graph of the corresponding Markov chain
$A_\mathbf{k}$	Arborescence with root $\mathbf{k}$
$\|A_\mathbf{k}\|$	Weight of an arborescence with root $\mathbf{k}$
q_e	Transition probability of edge $e \in E$ in a graph representing a Markov chain with nodes $\mathbf{k}$
γ	Euler-Mascheroni constant $\gamma = 0.5572$

Graph theory and matching

$G(V, E)$	Graph with (hyper-)edges $e \in E$ and vertices (nodes) $v \in V$
V_ι	Subset of nodes that represent the memories of party $\iota \in \{1, \ldots, N\}$
v_ι^j	Node $j \in \{1, 2, \ldots, m\}$ of party $\iota \in \{1, 2, \ldots, N\}$. Note that party $\iota = N$ is party A (i.e., $v_N^j = v_A^j$) and the nodes of parties $i = 1, \ldots, N-1$ are the nodes of the peers B_i
M	Matching: set of disjoint N-tuples with $\|M\| = \ell$
ζ	N-tuples contained in the matching: $\zeta_k \in M$ with $k \in \{1, \ldots, \|M\|\}$
$W(M)$	Weight of a matching M
s	Source of the graph given in network flow
t	Sink of the graph given in network flow
$u(v_1, v_2)$	Capacity of an edge (v_1, v_2)
f	Flow from source s to sink t in the network flow
l_max	Upper bound on the cardinality achievable in the multipartite matching for a given graph instance

(Multipartite) quantum key distribution

$\mathcal{B}_1, \mathcal{B}_2$	Orthogonal bases that are used to encode a classical bit with $\mathcal{B}_1 = \{\|0\rangle, \|1\rangle\}$ and $\mathcal{B}_2 = \{\|\tilde{0}\rangle, \|\tilde{1}\rangle\}$
n	Number of rounds the prepare-and-measure part of the QKD protocol is performed
Q_X	Quantum bit error rate in X-basis

Q_Z	Quantum bit error rate in Z-basis for the bipartite protocol
Q_{AB_i}	Bipartite quantum bit error rate in the multipartite protocol
$\langle X \rangle$	Expectation value of the X-operator
$\langle Z_A Z_{B_i} \rangle$	Expectation value of the Z-operator for party A and each B$_i$
r_∞	Asymptotic secret fraction
$h(p)$	Binary Shannon entropy function
K	Secret key rate
δ	Number of storage rounds of a qubit in the quantum memory
τ	Decoherence parameter of the quantum memory
ρ_ι	Werner state in Bell diagonal form that gives the initial state shared between each party $\iota \in \{1, 2, \ldots, N\}$ and the central quantum router
$\rho_{(x_1,x_2),\iota}$	Entry (x_1, x_2) of density matrix ρ of the ι-th party with x_1 indicating the row and x_2 the column of the entry
ρ_{tot}	Total initial state of all parties
ρ^{GHZ}	Output state: GHZ diagonal state after performing the entanglement swapping within the quantum router
$\lambda_0^\pm, \lambda_\kappa$	GHZ diagonal elements with $\kappa \in \{1, 2, \ldots, 2^{N-2}/2\}$
A, B, C, D	Bell diagonal elements
χ_{norm}	Normalization constant of ρ^{GHZ}

Paper A

On Matching in Multipartite Quantum Routers

Dagmar Bruß
Luis Gindorf
Julia Kunzelmann
dagmar.bruss@hhu.de
luis.gindorf@hhu.de
julia.kunzelmann@hhu.de
Heinrich-Heine-Universität Düsseldorf
MNF, Institut für Theoretische Physik III
Düsseldorf, NRW, Germany

Christian Laußmann
Jörg Rothe
christian.laussmann@hhu.de
rothe@hhu.de
Heinrich-Heine-Universität Düsseldorf
MNF, Institut für Informatik
Düsseldorf, NRW, Germany

Abstract

Over the past few years, the concept of quantum routers and their usefulness in quantum communication networks (e.g., in the BB84 protocol [5]) has been popularized in quantum information theory. While quite some work has been done on the theoretical implementation of quantum routers using multiplexing, most of it is constrained to the bipartite case [2, 9]. We extend this setup to quantum routers used in multipartite conference key agreement protocols so as to distribute a secret key among N parties. We formalize the general *quantum entanglement matching* problem, and show it's NP-completeness. We then study special cases for which we found efficient algorithms. Finally, we consider the weighted case where the weights represent the qubit ages.

Keywords

Quantum cryptography, Quantum routers, Computational complexity, Approximation, Graph theory, Matching

ACM Reference Format:

Dagmar Bruß, Luis Gindorf, Julia Kunzelmann, Christian Laußmann, and Jörg Rothe. 2025. On Matching in Multipartite Quantum Routers. In *Proceedings of (QUASAR '25)*. ACM, New York, NY, USA, 10 pages. https://doi.org/XXXXXXX.XXXXXXX

1 Introduction

Quantum communication, which emerged from quantum mechanics in the 20th century, studies communication via quantum information (qubits) over macroscopic distances [16]. It provides the advantage of an increased security against an eavesdropper since every interaction with a quantum system has an effect on it, and it is therefore more likely to detect any act of eavesdropping [5, 32].

This becomes especially important as many public-key cryptosystems are vulnerable to quantum algorithms (for more background on quantum cryptography, we refer to the survey by Bruß et al. [8]), although it is not clear yet whether quantum computers can be a real threat in practice [31]. The most common choice of transportation for the qubits is that of single photons in glass fiber. The polarization of the photon encodes the state of the qubit [34]. Their effective range, however, is limited due to photon losses via random scattering processes [30]. The transmittivity p of a photonic qubit in an optical fiber depends on the communication distance D:

$$p = 10^{-\frac{\alpha D}{10}}, \qquad (1)$$

where a reasonable assumption for the absorption coefficient α is 0.2 dB/km [16]. For a typical distance of $D = 100$ km, this yields a transition probability of $p = 0.01$, so the qubit arrives intact in only one percent of all cases.

To cope with this problem, the use of quantum repeaters—first without multiplexing [30] and later with multiplexing [9]—has been proposed. These network elements are placed between the communicating parties to shorten the distance between them. Links are first generated between two neighboring repeater stations. One long link between the end nodes is generated by performing measurements on the quantum repeaters. Here, "multiplexing" means that each party has several parallel quantum channels available, so several links between the parties and the quantum repeater can be produced simultaneously. Arriving qubits are stored in quantum memories within the router. These stored qubits are further used to generate entangled states between the N parties, where not only parallel links between the memories are considered. This is done to further increase the generation rate of links between end nodes. For the bipartite quantum repeater, there has been some research considering repeaters without [1, 7, 13, 22] and with multiplexing [2, 9]. Also, the generalization to the multipartite quantum repeater without multiplexing has been examined in [4, 11, 14, 33]. Often, multipartite quantum repeaters are called quantum routers because qubits are distributed more in a net-like topology than in a straight topology. As in the bipartite quantum repeater, the communication via quantum routers can be used for conference key agreement [24], the generalization of quantum key distribution. Here, qubits are sent to multiple other parties to share a secret key among all parties. So far, the generalization of multipartite multiplexing has not been analyzed much yet [20].

Due to the multiplexing, a matching protocol must be executed within the quantum router for every round of the quantum key distribution protocol. For the bipartite setup, the matching problem from graph theory is well-known, and there exist efficient algorithms to find a matching with maximum cardinality [6, 18].

QUASAR '25, Notre Dame, IN, USA

ACM ISBN 978-1-4503-XXXX-X/2018/06
https://doi.org/XXXXXXX.XXXXXXX

Multiple strategies to maximize the quantum repeater's efficiency have been implemented and analyzed in the bipartite case [2]. In the multipartite quantum router, the matching problem has first been considered in [20]. Our work further specifies the multipartite matching problem in the quantum router. We study the computational complexity of the associated problem and introduce efficient algorithms for some special cases.

2 Preliminaries

For a detailed overview of quantum information theory, we refer the reader to the textbook by Nielsen and Chuang [26].

2.1 The Bigger Picture

We consider a multipartite quantum network in which N parties want to share a secret key by performing the multipartite NBB84 quantum key distribution protocol [17], which generalizes the well-known bipartite BB84 quantum key distribution protocol [5]. Therefore, it is necessary for all parties to share entangled states (*genuine multipartite entanglement*), such as the GHZ state

$$|\psi\rangle = \frac{1}{\sqrt{2}}\left(|0\rangle^{\otimes N} + |1\rangle^{\otimes N}\right).$$

When the parties perform a measurement in the computational basis $\{|0\rangle, |1\rangle\}$ on this shared state, each party ideally records the same classical output of 0 or 1. Repeating this for n shared states, each party would hold the same bit string $x \in \{0,1\}^n$ that can be used as a key after performing some classical post-processing steps [17]. However, the distribution of entangled states is infeasible for long distances due to the exponential distance dependency of the transition probability p of a photon in a quantum channel (recall Equation (1)). To circumvent this, a quantum router is introduced to shorten the communication distances among the parties: Every party prepares several entangled qubit pairs and sends one of the qubits from each pair to the router. The router, in turn, acts on the qubits to entangle the parties' qubits. The quantum router further uses *multiplexing* to increase the key rate (i.e., the average fraction of secret key bits per sent qubit). Multiplexing means that every party sends multiple qubits in parallel—from which some may get lost in transition—and the router establishes entanglement among the received qubits.

2.2 Quantum Router

We formalize a *quantum router* as a graph $G = (V, E)$ with V partitioned into N parts. The nodes represent the filled quantum memories (of N communicating parties) in the quantum router.[1] One part (without loss of generality, the last one) is called the *center* and is denoted by $C = \{C^1, \dots\}$, and all other parts are called the *peers*, denoted by $P_i = \{P_i^1, \dots\}$ with $i \in [N-1]$, where $[k]$ denotes $\{1, \dots, k\}$ for $k \in \mathbb{N} \setminus \{0\}$. It holds that $V = P_1 \cup \cdots \cup P_{N-1} \cup C$; since this is a partition, all its parts are pairwise disjoint and cover all of V.

To generate an *entanglement link*, the quantum router performs a GHZ measurement (*entanglement swapping*) involving exactly one memory (i.e., node) from each part.[2] A GHZ measurement requires that one memory is designated as the *center*, as described in Figure 1. Multiple links can be created this way; however, each quantum memory is only allowed to appear in at most one entanglement link. Generally, quantum memories cannot be combined to an entanglement link arbitrarily. There can be limitations which memories can be connected due to the complicated physical realization (see Appendix A for details). To address this, we say an edge between two nodes in quantum router G represents the ability to use the respective nodes in an entanglement link; in other words, the edge represents the existence of a controlled-NOT (or, CNOT) gate. We assume that the center node in every entanglement link is always from the center C, which significantly reduces the difficulties in a physical realization. This brings us to the notion of *entanglement matching*, defined as follows.

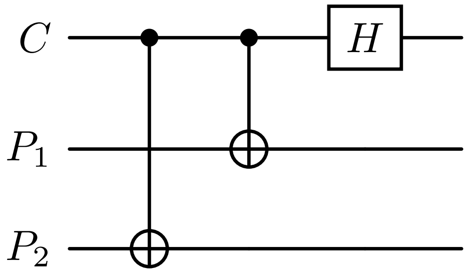


Figure 1: Quantum circuit for the GHZ state measurement for a tripartite network. The center party has a special role by providing the control qubit for the CNOT gate and additionally performing the Hadamard gate H.

Definition 2.1 (entanglement matching). Given a quantum router $G = (V, E)$ with V partitioned into N parts as explained above, an *entanglement matching* $M = \{\mu_1, \dots, \mu_q\}$ is a set of subsets $\mu_j \subseteq V$, $1 \le j \le q$, such that

(1) every subset μ_j in M consists of exactly one node from every part $P_1, \dots, P_{N-1}, C$,
(2) every two distinct subsets, say μ_i and μ_j with $i \neq j$, in M are disjoint, and
(3) every node x in any subset μ_j is either itself in the center C, or has an edge to the respective center node from μ_j.

The size of an entanglement matching M is its cardinality $|M|$.

Let us illustrate this with Figure 2.

2.3 Computional Complexity

We will study the computational complexity of decision problems related to quantum entanglement matchings. Assuming the reader to be familiar with the basic notions of computational complexity

[1] Note that in transmission some qubits may get lost, so the total number of nodes in the graph is not necessarily the total number of available memories. For simplicity, we do not consider empty memories, as they cannot be used anyway.

[2] This projects the locally held qubits of the parties (i.e., those qubits not sent to the router) onto a GHZ state from which the secret key can be generated [17]. However, this happens outside the router, so for the following it is not relevant.

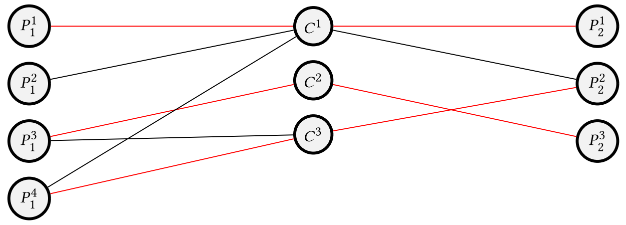


Figure 2: Example of a quantum router with three parts: the center $C = \{C^1, C^2, C^3\}$ and two peers, $P_1 = \{P_1^1, P_1^2, P_1^3, P_1^4\}$ and $P_2 = \{P_2^1, P_2^2, P_2^3\}$. A possible entanglement matching of size three is $M = \{\{C^1, P_1^1, P_2^1\}, \{C^2, P_1^3, P_2^3\}, \{C^3, P_1^4, P_2^2\}\}$ (in red).

theory, we refer to the standard textbooks by Garey and Johnson [15], Papadimitriou [27], and Rothe [28] for more background. In particular, we denote the complexity classes *"deterministic polynomial time"* by P and *"nondeterministic polynomial time"* by NP. Further, we consider the notions of *hardness* and *completeness for* NP, based on the *polynomial-time many-one reducibility*, denoted by $\leq_{\mathrm{m}}^{\mathrm{p}}$: For decision problems X and Y, we write $X \leq_{\mathrm{m}}^{\mathrm{p}} Y$ if and only if every instance x of X can be transformed by a polynomial-time computable reduction ρ to an instance $\rho(x)$ of Y such that x is a yes-instance of X exactly if $\rho(x)$ is a yes-instance of Y. A problem Y is said to be NP-*hard* if $X \leq_{\mathrm{m}}^{\mathrm{p}} Y$ for each X in NP, and Y is said to be NP-*complete* if it is both NP-hard and in NP.

3 Computional Complexity of the Quantum Entanglement Matching Problem

To study the computional complexity of determining whether there exist large enough quantum entanglement matchings for a given quantum router, we first formalize this as a decision problem:

Quantum Entanglement Matching (QEM)	
Given:	A quantum router G with center C and peers $P_1, \ldots, P_{N-1}$, and a positive integer γ.
Question:	Does there exist an entanglement matching of size at least γ in G?

Note that the problem is different from the NP-complete *N-dimensional matching problem* [23], where a matching is requested for given hyper-edges. Our graph (representing a quantum router) is a regular graph without hyper-edges. We will now show that QEM is NP-complete, too.

Theorem 3.1. *QEM is* NP*-complete.*

Proof. Membership in NP is obvious since given an instance of QEM, in polynomial time we can nondeterministically guess a solution and deterministically check its correctness, i.e., for each guessed entanglement matching, we can easily verify its size and validity.

It remains to show NP-hardness, which we will do now by a reduction to QEM from the well-known NP-complete problem 3SAT:[3]

3-Satisfiability (3SAT)	
Given:	A boolean formula π in conjunctive normal form with exactly three literals per clause.
Question:	Is π satisfiable, i.e., can we assign truth values to its variables such that the formula evaluates to true?

Let

$$\pi = \left(\pi_1^1 \vee \pi_1^2 \vee \pi_1^3\right) \wedge \cdots \wedge \left(\pi_n^1 \vee \pi_n^2 \vee \pi_n^3\right)$$

be a given instance of 3SAT, where each π_i^j, $1 \leq i \leq n$ and $1 \leq j \leq 3$, is a literal over the set $\alpha = \{\alpha_1, \ldots, \alpha_q\}$ of variables. By $\overline{\alpha} = \{\neg\alpha_1, \ldots, \neg\alpha_q\}$ we denote the set of negations of these variables, so each π_i^j is from $\alpha \cup \overline{\alpha}$.

Given π, we construct a quantum router $G = (V, E)$ with

$$V = C \cup P_1 \cup \cdots \cup P_n \cup X,$$

where

$$\begin{aligned} C &= \alpha \cup \overline{\alpha} \cup \{\omega\} \text{ (with } \omega \text{ standing for "the last resort")}, \\ P_i &= \{P_i^{\alpha_1}, \ldots, P_i^{\alpha_q}, P_i^{\pi_i}\} \text{ for each } i,\ 1 \leq i \leq n, \text{ and} \\ X &= \{x^{\alpha_1}, \ldots, x^{\alpha_q}, x^{\omega}\}. \end{aligned}$$

Furthermore, the edges in E are defined as follows:

- For each x^{α_i}, there is an edge to α_i and $\neg\alpha_i$;
- for x^{ω}, there is an edge to ω;
- for each $P_i^{\alpha_r}$ with $r \in \{1, \ldots, q\}$ and $i \in \{1, \ldots, n\}$, there is an edge to α_r, $\neg\alpha_r$, and ω,
- for each $P_i^{\pi_i}$ with $i \in \{1, \ldots, n\}$, there is an edge to π_i^1, π_i^2, and π_i^3 according to the given formula $\pi = \left(\pi_1^1 \vee \pi_1^2 \vee \pi_1^3\right) \wedge \cdots \wedge \left(\pi_n^1 \vee \pi_n^2 \vee \pi_n^3\right)$.

We illustrate the construction in Figure 3 with a simple 3SAT instance. Note that the constructed graph meets the requirements of a quantum router matching. To complete our construction of the QEM problem instance, we set $\gamma = q + 1$.

Intuitively, the selection of nodes from the center C in the matching will represent a truth assignment to the variables in π, and the peers $P_1, \ldots, P_n$ represent the clauses in π. Note that, by construction, a matching of size q is always possible since we can match every α_j with x^{α_j} and $P_i^{\alpha_j}$ for every P_i. Note further that the construction can be performed in polynomial time.

It remains to show the correctness of the construction: To prove 3SAT $\leq_{\mathrm{m}}^{\mathrm{p}}$ QEM, we need to show that for each given boolean formula π in conjunctive normal form with exactly three literals per clause, it holds that

$$\pi \in \text{3SAT} \iff (G, \gamma) \in \text{QEM}.$$

($\Rightarrow$) We first show that if there exists a truth assignment to the variables that makes π true, we can find a quantum router matching of size γ.

Let $\beta_1, \ldots, \beta_q \in \alpha \cup \overline{\alpha}$ be chosen such that the β_i represent the given satisfying truth assignment: $\beta_j = \alpha_j$ if α_j is set to true, and $\beta_j = \neg\alpha_j$ otherwise. Define a quantum entanglement matching M by constructing the following subsets of M:

- For each β_j, $1 \leq j \leq q$, we construct a subset with β_j (from the center) and all adjacent, not already used $P_i^{\pi_i}$. For each P_i such that $P_i^{\pi_i}$ is not adjacent to β_j or has already been used

[3] Cook [10] and, independently, Levin [21] proved that 3SAT is NP-complete; it is the first natural problem known to be NP-complete, see also the relevant textbooks [15, 27, 28] mentioned above.

earlier, we add $P_i^{\alpha_j}$ instead. Finally, we add x^{α_j} to complete the subset. Note that every $P_i^{\pi_i}$ is now contained in exactly one of the subsets constructed in this step, since each $P_i^{\pi_i}$ has an edge to at least one β_j because the assignment satisfies π.

- For node ω, we do the same, except that this will not match with any $P_i^{\pi_i}$, as there is no connection to ω, and they are already used in subsets constructed in the previous step. However, since in each P_i, node $P_i^{\pi_i}$ was matched instead of one $P_i^{\alpha_j}$, ω can be matched with these leftovers as well as with x^{ω}.

The resulting matching will have size $\gamma = q + 1$, since all nodes from C (including ω) are matched. Further, the matching is valid since (1) exactly one node from $C, X, P_1, \ldots, P_n$ is part of the matching, (2) no node is selected twice, and (3) only nodes adjacent to the center can be selected.

($\Leftarrow$) We now show that if there is a matching of size $\gamma = q + 1$, then there is also a truth assignment making π true. Let $M = \{\mu_1, \ldots, \mu_\gamma\}$ be a quantum router matching of size (at least) γ. First, note that subset X controls that α_i and $\neg\alpha_i$ can never be selected in the same matching, as only one of the two can find a peer in X. Thus M has exactly size γ, and it contains either α_i or $\neg\alpha_i$ for each i, $1 \leq i \leq q$, as well as ω. This in turn means that the selected literals (or negations) in M are a valid (potentially not satisfying) truth assignment to the variables of π. It only remains to show that the selection actually satisfies π.

For a contradiction, assume this truth assignment does not satisfy π. This means that at least one clause of $(\pi_i^1 \vee \pi_i^2 \vee \pi_i^3)$ is not satisfied by the truth assignment. This implies that node $P_i^{\pi_i}$ is not part of the matching M, as it only has edges to the respective variables that satisfy this clause. But this is a contradiction to the size of M being $\gamma = q + 1$, since only q nodes remain in P_i. Thus the truth assignment must satisfy π, proving the claim. □

As we see, finding the largest possible matching can be quite challenging and becomes intractable on large instances. However, in practical applications, some assumptions can be made on the set of edges E in the graph from a QEM instance that allow for an efficient solvability of the problem. Let us first consider a case where each node is connected to at most one node from the center.

Theorem 3.2. *QEM is in* P *whenever every node from the peers is connected to at most one node from the center.*

Proof. We give an algorithm that finds a maximum matching for a given instance in polynomial time. For each node from the center, we check if it has at least one connection to a node from every peer. If so, we choose one arbitrary node from each peer which is connected to this center node, add them to the matching (together with the center node). It is easy to see that this can be done in polynomial time (more precisely, in time $O(N \cdot r)$, where N is the size of the partition and r the number of center nodes).

Note that since each node has at most one connection to a node from the center, no node will be used twice by the algorithm. Thus all subsets in the matching are disjoint. Further, by construction, only nodes connected to the respective center node are considered, and each subset in the matching has exactly one node from every peer and from the center. Thus the matching is valid.

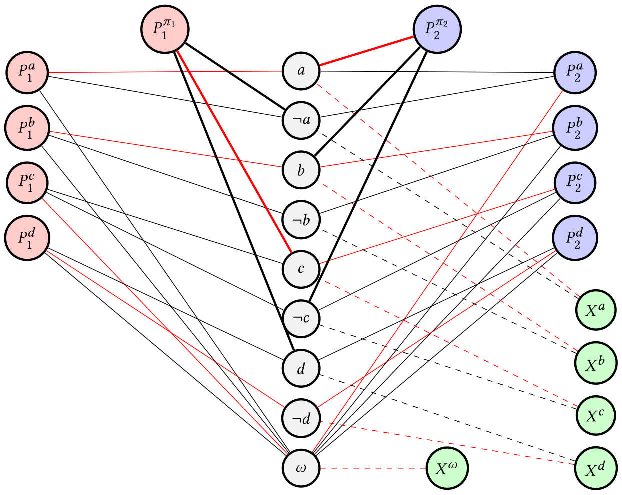


Figure 3: Illustration of the construction in the proof of Theorem 1. The shown graph represents the 3SAT formula $\pi = \pi_1 \wedge \pi_2 = (\neg a \vee c \vee d) \wedge (a \vee b \vee \neg c)$. The four parts are color-encoded. A possible quantum entanglement matching is indicated in red and corresponds to the assignment $a, b, c, \neg d$, which indeed satisfies π.

Finally, the resulting matching is maximal, as every center node appears in a subset of the matching, except if it does not have a connection to at least one other party. In the latter case, the respective center node cannot be part of the matching anyway.

By applying this algorithm we can easily decide whether a matching of the requested size at least γ is possible. □

We next consider a case where every node outside the center is connected to every node within the center. That is, we have the following set of edges:

$$E = \{\{x, y\} \mid x \in C,\ y \in V \setminus C\}.$$

Theorem 3.3. *QEM is in* P *whenever every node from the peers is connected to every node within the center.*

Proof. We give an algorithm that finds a maximum matching for a given instance in $O(N \cdot \ell)$, where N is the size of the partition and ℓ the minimum number of nodes in any of its parts: Choose one arbitrary node from each part and add them to the matching. Remove all chosen nodes to ensure they are not reused. Iterate this until one of the parts is empty (exactly ℓ times).

The resulting matching is maximal, as no more nodes from the now empty part are available. Further, it is valid since, by construction, each subset μ_i in the matching

(1) contains exactly one node from every part of the partition,
(2) is disjoint to every other subset μ_j, $j \neq i$, as we remove used nodes from the graph, and
(3) nodes within μ_i are guaranteed to have a connection to their respective center (or are themselves in the center).

By applying this algorithm, we can easily decide whether a matching of the requested size at least γ is possible. □

Interestingly, we now show that there exists an efficient algorithm for QEM whenever $N \leq 3$, i.e., when a center is surrounded by at most two peers in the given quantum router. This underlines a fundamental difference between QEM and the N-dimensional matching problem, which is NP-hard even for three dimensions [15, 19].

Theorem 3.4. *QEM is in* P *for instances with* $N \leq 3$*, i.e., with one center and at most two peers.*

Proof Sketch. For $N = 1$ there is no peer to match with the center, so this is a trivial case. For $N = 2$ the problem reduces to the regular bipartite matching problem, for which an efficient algorithm exists [18].

It turns out that the case $N = 3$ can be formulated as a *network flow problem.* In general, a *network flow* is defined as follows (see, e.g., the textbook by Ahuja et al. [3]): Let $G(V, E)$ be a directed graph with the distinguished nodes $s, t \in V$ as the source and sink. The capacity $c(e) \in \mathbb{R}$ of an edge $e \in E$ defines the maximum flow that can go through that edge. The flow $f : E \longrightarrow \mathbb{R}$ has to fulfill the following properties:

a) For each edge e holds $f(e) \leq c(e)$.
b) For each node, the sum of all input flow equals the sum of all output flow.

The question in network flow is: what is the maximum flow f_{max} that can be achieved in a given instance?

To model QEM as a network flow problem, we first introduce a source s and a sink t. We add directed edges from s to every node in P_1, and directed edges from every node in P_2 to t. We then replace all undirected edges between P_1 and C by directed edges from P_1 to C. Next, we clone every center node C^i and denote the clone by $C^{i'}$. We introduce directed edges from every C^i to $C^{i'}$. Further, for each undirected edge between C^i and a node from P_2, we introduce a directed edge from $C^{i'}$ to the respective node from P_2. Finally, we remove all undirected edges and set the capacity of all remaining edges to 1. An example is given in Figure 4.

Note that each node from P_1 has exactly one incoming edge, and nodes from C and P_2 have exactly one outgoing edge, and all edges have capacity 1. This mimics the QEM-requirement that every node is part of at most one entanglement link (see Definition 2.1 (2)). Note further that to get from source to sink one has to pass *exactly* one node from C, P_1, and P_2. This represents the requirement from Definition 2.1 (1). Finally, note that Definition 2.1 (3) is guaranteed by the construction since edges to and from the center (resp. center-clone) are only turned from undirected to directed but never introduced or removed.

To complete our proof sketch, note that a maximum flow uses the maximum number of paths from the source to the sink, since all weights are 1. Hence, the matching is maximal. □

The restrictions on QEM introduced previously demonstrate that efficient algorithms exist for specific graph structures. A quantum router where every node has at most one connection to the center (see Theorem 3.2) is relatively easy to realize in practice, and it is helpful to know that it is also easy to find optimal entanglement matchings in such a router. However, it will not be possible to match as many quantum memories as in the fully connected router described in Theorem 3.3.[4] While this case is also easy from a computational perspective, it may in contrast be physically challenging to implement. The last case with $N \leq 3$ is interesting because when it comes to experimental realization of multipartite quantum routers, the case of three parties will be the first proof of concept. It is good to know that in such a proof of concept we at least will not run into computational intractability.

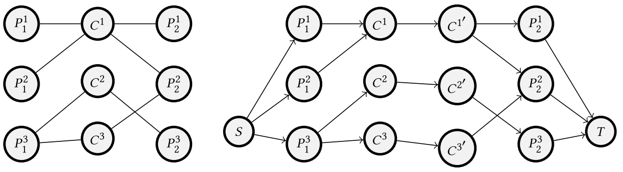


Figure 4: Left: example of a quantum router with three parties. Right: The same quantum router mapped to a network flow instance. The capacity of every edge is 1.

4 Weighted Matchings

We now conclude this study by adding weight to the matchings. These weights represent the qubits' storage time in the quantum memories. We call this the *qubit age*, denoted by $\delta_v \in \mathbb{N}$ for a node v. The background for this is that, in reality, one would repeat the process of entanglement matching multiple times until a sufficiently large key was exchanged. Between the single rounds of this process, the unmatched filled memories will not be cleared. However, during storage the qubits decohere in the quantum memories. The more a qubit decoheres, the less useful it is for entanglement swapping. Informally, what we try to do in this section is to find a cardinality-maximal entanglement matching (as before in the unweighted versions) which simultaneously maximizes the usability of the involved qubits for entanglement swapping. To do so, we first define this *usability of qubits.* We then define the weighted version of QEM formally and study its computational complexity.

The quality of a qubit is given by the *fidelity*, which is 1 initially and decreases when the qubit decoheres over time:

$$F(v) = \exp\left(-\delta_v/\tau\right) \tag{2}$$

with the decoherence constant τ, which is specific to the quantum memory. The qubit age is measured in entanglement generation attempts (rounds). Note that this is based on a specific error model. Here, we assume that qubits depolarize in the quantum memory (see Appendix A for more details). Due to decoherence, errors arise that diminish the ratio of secret bits generated in the BB84 protocol [5] from those imperfect qubits. The quantum bit error rates (QBERs) give the errors after performing the entanglement swapping. They depend on the storage time of the qubits before the entanglement swapping. We distinguish between the error rate e_X measured in the X-basis and the bipartite error rate e_{CP_i} between the parties C

[4] According to Abruzzo et al. [2] and Kunzelmann et al. [20], the advantage of all-to-all connections compared to more sparse connections does not seem to be overwhelming, though.

and P_i measured in the Z-basis:

$$e_X = \frac{1}{2} - \frac{1}{2 \cdot 3^N} \prod_{k=1}^{N} (4F_k - 1); \tag{3}$$

$$e_{CP_i} = \frac{2}{9} \left(F_C + F_{P_i} - 4F_C F_{P_i} + 2 \right). \tag{4}$$

The general goal is to optimize the secret fraction. It is given by:

$$r_\infty \left(\bar{e}_X, \bar{e}_{CP_i} \right) = \max \left\{ 0, 1 - h(\bar{e}_X) - \max_i h(\bar{e}_{CP_i}) \right\} \tag{5}$$

with binary Shannon entropy

$$h(p) = -p \log_2(p) - (1-p) \log_2(1-p).$$

Note that the QBERs $(\bar{e}_X, \bar{e}_{CP_i})$ are here averaged over one matching. The overall key rate is given by the fraction of all shared bits (given as the router rate R) from which a secret key can be generated:

$$K = r_\infty \left(\bar{e}_X, \bar{e}_{CP_i} \right) R. \tag{6}$$

The router rate is proportional to the number of shared entangled links generated per round:

$$R \propto |M|. \tag{7}$$

We provide a detailed description of the underlying physical background of the entanglement swapping protocol as well as the derivation of Equations (3) and (4) in Appendix A. Given the ages of all qubits, we now optimize for the secret fraction r_∞ (recall Equation (5)) while keeping the requirement of maximum cardinality. We first observe the following result:

Theorem 4.1. *Younger qubits lead to a greater secret fraction if all involved qubits are younger than $\tau \log(4)$. Formally:*

$$\forall i \in [N], j \in [m] : \delta_{ij} < \tau \log(4) \implies \forall i \in [N], j \in [m] : \frac{\partial r_\infty}{\partial \delta_{ij}} < 0.$$

Proof. We begin by deriving two statements from the condition $\delta_{ij} < \tau \log(4)$.

As a shorthand, we define $F_{ij} := F(\delta_{ij}) = \exp(-\delta_{ij}/\tau)$. As F is monotonically decreasing in δ, we have:

$$\delta_{ij} \in [0, \log(4)\tau[\implies F_{ij} \in]F(\log(4)\tau), F(0)] = \left] \frac{1}{4}, 1 \right] \tag{8}$$

From this, we can deduce bounds for the QBERs:

$$e_X(j) \geq \min \left\{ \frac{1}{2} - \frac{1}{2 \cdot 3^N} \prod_{i=1}^{N} (4F_{ij} - 1) \middle| F_{ij} \in]1/4, 1] \right\} = 0;$$

$$e_X(j) \leq \max \left\{ \frac{1}{2} - \frac{1}{2 \cdot 3^N} \prod_{i=1}^{N} (4F_{ij} - 1) \middle| F_{ij} \in]1/4, 1] \right\} < \frac{1}{2}.$$

And similarly for e_{CP_i}:

$$e_{CP_i} \geq \min \left\{ \frac{2}{9}(F_{1j} + F_{ij} - 4F_{1j}F_{ij} + 2 \middle| F_{ij} \in]1/4, 1] \right\} = 0;$$

$$e_{CP_i} \leq \max \left\{ \frac{2}{9}(F_{1j} + F_{ij} - 4F_{1j}F_{ij} + 2 \middle| F_{ij} \in]1/4, 1] \right\} < \frac{1}{2}.$$

Since the averaged QBERs are calculated by taking the mean of the QBERs, we receive the same bounds:

$$\bar{e}_X \in [0, 1/2[, \quad \bar{e}_{CP_i} \in [0, 1/2[$$

Now, we can infer the second statement:

$$h(e) = -e \log_2(e) - (1-e) \log_2(1-e)$$
$$\implies \frac{\partial h(e)}{\partial e} = \log_2 \left(\frac{1}{e} - 1 \right) > 0 \text{ for } e \in \left[0, \tfrac{1}{2} \right[\tag{9}$$

We now complete the proof of the theorem by derivation:

$$\frac{\partial r_\infty}{\partial \delta_{ij}} = \frac{\partial}{\partial \delta_{ij}} \max \left\{ 0, 1 - h(\bar{e}_X) - \max_{k \in [N-1]} h(\bar{e}_{CP_k}) \right\}.$$

We first consider the case $1 - h(\bar{e}_X) - \max_{k \in [N-1]} h(\bar{e}_{CP_k}) \leq 0$. Here, $r_\infty \equiv 0$ and thus $\frac{\partial r_\infty}{\partial \delta_{ij}} = 0$. Of course, the other case is more interesting:

$$\begin{aligned} \frac{\partial r_\infty}{\partial \delta_{ij}} &= -\frac{\partial}{\partial \delta_{ij}} h(\bar{e}_X) - \frac{\partial}{\partial \delta_{ij}} \max_{k \in [N-1]} h(\bar{e}_{CP_k}) \\ &= -\frac{\partial h(e)}{\partial e} \left(\frac{\partial \bar{e}_X}{\partial \delta_{ij}} + \frac{\partial \bar{e}_{CP_p}}{\partial \delta_{ij}} \right) \\ &= -\frac{\partial h}{\partial e} \frac{1}{l} \frac{\partial}{\partial \delta_{ij}} \left(\sum_{r \in [l]} e_X(r) + \sum_{r \in [l]} e_{CP_p}(r) \right), \end{aligned}$$

where p (i.e., the subscript of party P_p, occurring in the second and the third equality) is set to $\operatorname{argmax}_k h(\bar{e}_{CP_k})$.

Since $e_X(r)$ depends on δ_{ir} for all $i \in [N]$, only the $e_X(j)$-term is dependent on δ_{ij} and survives the derivation.

Further, $e_{CP_p}(r)$ depends on δ_{1r} and δ_{pr}. Therefore, the sum collapses with $r = j$ through this derivation, too:

$$\begin{aligned} \frac{\partial r_\infty}{\partial \delta_{ij}} &= -\frac{\partial h}{\partial e} \frac{1}{l} \frac{\partial}{\partial \delta_{ij}} \left(e_X(j) + e_{CP_p}(j) \right) \\ &= -\frac{1}{l} \frac{\partial h}{\partial e} \frac{\partial F_{ij}}{\partial \delta_{ij}} \frac{\partial}{\partial F_{ij}} \left(e_X(j) + e_{CP_p}(j) \right) \quad \Big| \ \frac{\partial F_{ij}}{\partial \delta_{ij}} = -\frac{F_{ij}}{\tau} \\ &= \frac{F_{ij}}{l\tau} \frac{\partial h}{\partial e} \frac{\partial}{\partial F_{ij}} \left(e_X(j) + e_{CP_p}(j) \right) \quad \Big| \ C := \frac{F_{ij}}{l\tau} \frac{\partial h}{\partial e} \\ &= C \left(\frac{\partial e_X(j)}{\partial F_{ij}} + \frac{\partial e_{CP_p}(j)}{\partial F_{ij}} \right). \end{aligned}$$

We find $C > 0$ due to statements (8) and (9). Also using (8), we calculate

$$\begin{aligned} \frac{\partial e_X(j)}{\partial F_{ij}} &= -\frac{1}{2 \cdot 3^N} \prod_{t \neq i} (4F_{tj} - 1) \frac{\partial}{\partial F_{ij}} (4F_{ij} - 1) \\ &= -\frac{2}{3} \prod_{t \neq i} \frac{4F_{tj} - 1}{3} < 0; \end{aligned}$$

$$\frac{\partial e_{CP_p}(j)}{\partial F_{ij}} = \left\{ \begin{matrix} \frac{2}{9}(1 - 4F_{pj} & i = 1 \\ \frac{2}{9}(1 - 4F_{1j} & i = p \\ 0 & \text{else} \end{matrix} \right\} \leq 0.$$

Putting this together yields $\frac{\partial r_\infty}{\partial \delta_{ij}} < 0$, completing the proof. □

We emphasize that the qubit age condition of the theorem is quite weak. Qubits older than $\log(4)\tau$ lead to quantum bit error rates greater than 1/2, making their use for the protocol worse than randomly guessing qubit states. Therefore, qubits older than this threshold might as well be just discarded.

Theorem 4.1 tells us to use the ℓ youngest available qubits for an optimal matching of cardinality ℓ. It is, however, still unclear in which way qubits should be combined in order to maximize the secret fraction. Because the secret fraction is a highly nonlinear function with respect to the qubit ages, we introduce a simpler optimization goal:

Postulate 1. *An entanglement matching M has the maximal possible secret fraction for an instance if it minimizes*

$$\xi(M) = \sum_{\mu_j \in M} \prod_{x \in \mu_j} \delta_x .$$

We justify this postulate by looking at the expression of the QBERs in Equations (3) and (4). For an entanglement link, they are small when the product of the fidelities of the qubits involved is large. Therefore, the simplified optimization goal minimizes the product over the involved qubit's ages, since small qubit ages lead to high fidelities. For multiple matches in a matching, we sum these products, since the QBERs are averaged in this case. Minimization of the qubit ages also aligns with the statement of Theorem 4.1. We can now define the weighted version of the Quantum Entanglement Matching problem:

Weighted Quantum Entanglement Maximum Matching (WQEMM)

Given:	A quantum router G with center C and peers $P_1, \dots, P_{N-1}$, an age $\delta_v \in \mathbb{N}$ for each node $v \in V$, and a positive integer γ.
Question:	Does there exist a maximum-cardinality entanglement matching M with $\xi(M) \leq \gamma$ in G?

It is easy to see that for the general case (i.e., an arbitrary graph G) this problem is NP-hard, since finding an *unweighted* maximum-cardinality matching is already NP-hard by Theorem 3.1. However, for two of the aforementioned special cases—nodes connected to at most one node from the center, or to all nodes from the center—we now present efficient algorithms for WQEMM, thus strengthening Theorems 3.2 and 3.3.

Theorem 4.2. *WQEMM is in* P *whenever every peer's node is connected to at most one node from the center.*

Proof. We extend the algorithm from the unweighted case in Section 3 so that it not only finds a maximum-cardinality matching M but also minimizes $\xi(M)$.

For each node $c \in C$ from the center, we check if it has at least one connection to a node from every peer. If the respective center node has not, we skip it. If it has, let $P_i[c]$ be the set of nodes in P_i connected to center node c. We choose from each peer $i \in \{1, \dots, N-1\}$ one node from $P_i[c]$ with the smallest storage time among the nodes in $P_i[c]$, entangle them with c, and add this entangled link to the matching.

It is easy to see that this can be done in polynomial time and leads to a valid maximum-cardinality matching (just as in the unweighted case). This matching also minimizes $\xi(M)$, due to the fact that every peer's node has only one connection to a center node: It is always better to choose a node of smallest age from $P_i[c]$, as the nodes in $P_i[c]$ cannot be used with any other center node, and choosing one of smallest age minimizes for the current entanglement link. By applying this algorithm one can decide WQEMM. □

Theorem 4.3. *WQEMM is in* P *whenever every node outside the center is connected to every node within the center.*

Proof. We give an algorithm that finds an optimal matching in time $O(m \log(m) N)$:

- Sort the nodes of every part of the partition by ascending weight (qubit age).
- Let ℓ be the minimum number of nodes in all parts of the partition (note that this is the maximal cardinality any matching can have).
- For every $k \in [\ell]$, add the k-th node from every peer and the center as an entanglement link to the matching.

It is clear that the produced matching is valid:

- Every match contains a node from every part of the partition.
- No two matches use the same node.
- Connectivity to the center is always guaranteed in this special case of the problem.

The matching also has maximal cardinality (i.e., ℓ). It remains to show that M is also optimal in

$$\xi(M) = \sum_{\mu_j \in M} \prod_{x \in \mu_j} \delta_x .$$

This is an application of the generalized rearrangement inequality [29, Theorem I], which states that for an arbitrarily shaped matrix of positive real numbers, the sum over the product of the columns is maximal when the matrix is permuted such that the columns are in nonincreasing order. In our case, it is minimal when the columns are in nondecreasing order. By applying this algorithm one can decide WQEMM in polynomial time. □

5 Conclusions and Outlook

We have studied computational aspects of quantum routers and of the problem of finding entanglement matchings. We have shown that the general case is NP-complete (Theorem 3.1) and explored special cases which are tractable and allow efficient algorithms (Theorems 3.2, 3.3, and 3.4). We then studied the weighted version of the problem (i.e., with qubit ages). Under a few reasonable assumptions we were able to prove the analogues of Theorems 3.2 and 3.3 for the weighted case (Theorems 4.2 and 4.3), i.e., these two weighted special cases are tractable, too.

As a next step, we already started with the design of efficient approximation algorithms for the general problem. We plan to study those algorithms both from a theoretical and experimental angle (i.e., worst case, best case, and average quality of the result). As we have shown that the general case is computationally hard, good approximation is one key to making quantum routers more realizable in practice.

Moreover, modern SAT-solving techniques may provide the possibility to circumvent the NP-completeness of QEM and WQEMM in practice. In addition, it would be interesting to see if our NP-complete problems QEM and WQEMM perhaps are *fixed-parameter tractable* [25] or whether they are hard also in terms of *parameterized* complexity [12] for reasonable parameters.

Acknowledgments

This work was supported in part by the Federal Ministry of Education and Research (BMBF) via the project QR.X (16KISQ023) and by Deutsche Forschungsgemeinschaft under DFG grant RO-1202/21-2 (project number 438204498) and under Germany's Excellence Strategy – Cluster of Excellence "Matter and Light for Quantum Computing" (ML4Q) EXC 2004/1-390534769.

References

[1] Silvestre Abruzzo, Hermann Kampermann, and Dagmar Bruß. 2014. Measurement-Device-Independent Quantum Key Distribution with Quantum Memories. *Physical Review A* 89, 1 (Jan. 2014), 012301. doi:10.1103/PhysRevA.89.012301

[2] Silvestre Abruzzo, Hermann Kampermann, and Dagmar Bruß. 2014. Finite-Range Multiplexing Enhances Quantum Key Distribution via Quantum Repeaters. *Physical Review A* 89, 1 (2014), 012303. doi:10.1103/PhysRevA.89.012303

[3] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. 1993. *Network Flows: Theory, Algorithms, and Applications.* Prentice Hall, Hoboken, NJ, USA.

[4] Guus Avis, Filip Rozpedek, and Stephanie Wehner. 2023. Analysis of Multipartite Entanglement Distribution Using a Central Quantum-Network Node. *Physical Review A* 107 (Jan 2023), 012609. Issue 1. doi:10.1103/PhysRevA.107.012609

[5] Charles H. Bennett and Gilles Brassard. 1984. Quantum Cryptography: Public Key Distribution and Coin Tossing. In *Proceedings of the IEEE International Conference on Computers, Systems, and Signal Processing*. IEEE Press, New York, NY, USA, 175–179.

[6] Claude Berge. 1957. Two Theorems in Graph Theory. *Proceedings of the National Academy of Sciences* 43, 9 (1957), 842–844.

[7] Hans-J. Briegel, Wolfgang Dür, J. Ignacio Cirac, and Peter Zoller. 1998. Quantum Repeaters: The Role of Imperfect Local Operations in Quantum Communication. *Physical Review Letters* 81 (Dec. 1998), 5932–5935. Issue 26. doi:10.1103/PhysRevLett.81.5932

[8] Dagmar Bruß, Gábor Erdélyi, Tim Meyer, Tobias Riege, and Jörg Rothe. 2007. Quantum Cryptography: A Survey. *Comput. Surveys* 39, 2 (2007), article 6, 27 pp.

[9] Odell A. Collins, Stewart David Jenkins, Alex Kuzmich, and T. A. Brian Kennedy. 2007. Multiplexed Memory-Insensitive Quantum Repeaters. *Physical Review Letters* 98 (2007), 060502. Issue 6.

[10] Stephen A. Cook. 1971. The Complexity of Theorem-Proving Procedures. In *Proceedings of the 3rd ACM Symposium on Theory of Computing*. ACM Press, New York, NY, USA, 151–158.

[11] Tim Coopmans, Robert Knegjens, Axel Dahlberg, David Maier, Loek Nijsten, Julio de Oliveira Filho, Martijn Papendrecht, Julian Rabbie, Filip Rozpedek, Matthew Skrzypczyk, Leon Wubben, Walter de Jong, Damian Podareanu, Ariana Torres-Knoop, David Elkouss, and Stephanie Wehner. 2021. NetSquid, a NETwork Simulator for QUantum Information Using Discrete Events. *Communications Physics* 4, 1, Article 164 (2021), 15 pages. doi:10.1038/s42005-021-00647-8

[12] Rodney G. Downey and Michael R. Fellows. 2013. *Parameterized Complexity* (2nd ed.). Springer-Verlag, Heidelberg and Berlin, Germany.

[13] Wolfgang Dür, Hans-J. Briegel, J. Ignacio Cirac, and Peter Zoller. 1999. Quantum Repeaters Based on Entanglement Purification. *Physical Review A* 59 (Jan. 1999), 169–181. Issue 1. doi:10.1103/PhysRevA.59.169

[14] Michael Epping, Hermann Kampermann, and Dagmar Bruß. 2016. Large-Scale Quantum Networks Based on Graphs. *New Journal of Physics* 18, 5 (May 2016), 053036. doi:10.1088/1367-2630/18/5/053036

[15] Michael Garey and David S. Johnson. 1979. *Computers and Intractability: A Guide to the Theory of NP-Completeness.* W. H. Freeman and Company, New York, NY, USA.

[16] Nicolas Gisin, Grégoire Ribordy, Wolfgang Tittel, and Hugo Zbinden. 2002. Quantum Cryptography. *Reviews of Modern Physics* 74 (2002), 145–195. Issue 1.

[17] Federico Grasselli, Hermann Kampermann, and Dagmar Bruß. 2018. Finite-Key Effects in Multipartite Quantum Key Distribution Protocols. *New Journal of Physics* 20, 11 (2018), 113014. doi:10.1088/1367-2630/aaec34

[18] John E. Hopcroft and Richard M. Karp. 1973. An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM J. Comput.* 2, 4 (1973), 225–231. doi:10.1137/0202019

[19] Richard M. Karp. 1972. Reducibility among Combinatorial Problems. In *Complexity of Computer Computations*, Raymond E. Miller, James W. Thatcher, and Jean D. Bohlinger (Eds.). Plenum Press, New York, NY, USA, 85–103.

[20] Julia A. Kunzelmann, Hermann Kampermann, and Dagmar Bruß. 2024. Multipartite multiplexing strategies for quantum routers. *Physical Review A* 110 (Sept. 2024), 032617. Issue 3. doi:10.1103/PhysRevA.110.032617

[21] Leonid Levin. 1973. Universal Sorting Problems. *Problemy Peredaci Informacii* 9 (1973), 115–116. In Russian. English translation in *Problems of Information Transmission*, 9:265–266, 1973.

[22] Zheng-Da Li, Rui Zhang, Xu-Fei Yin, Li-Zheng Liu, Yi Hu, Yu-Qiang Fang, Yue-Yang Fei, Xiao Jiang, Jun Zhang, Li Li, Nai-Le Liu, Feihu Xu, Yu-Ao Chen, and Jian-Wei Pan. 2019. Experimental Quantum Repeater Without Quantum Memory. *Nature Photonics* 13, 9 (2019), 644–648. doi:10.1038/s41566-019-0468-5

[23] László Lovász and Michael D. Plummer. 2009. *Matching Theory.* Vol. 367. AMS Chelsea Publishing, New York, NY, USA.

[24] Gláucia Murta, Federico Grasselli, Hermann Kampermann, and Dagmar Bruß. 2020. Quantum Conference Key Agreement: A Review. *Advanced Quantum Technologies* 3, 11 (2020), 2000025.

[25] Rolf Niedermeier. 2006. *Invitation to Fixed-Parameter Algorithms.* Oxford University Press, Oxford, UK.

[26] Michael A. Nielsen and Isaac L. Chuang. 2010. *Quantum Computation and Quantum Information.* Cambridge University Press, Cambridge, UK.

[27] Christos Papadimitriou. 1995. *Computational Complexity* (second ed.). Addison-Wesley, Reading, MA, USA.

[28] Jörg Rothe. 2005. *Complexity Theory and Cryptology. An Introduction to Cryptocomplexity.* Springer-Verlag, Heidelberg and Berlin, Germany.

[29] H. D. Ruderman. 1952. Two New Inequalities. *The American Mathematical Monthly* 59, 1 (1952), 29–32. https://www.jstor.org/stable/2307185

[30] Valerio Scarani, Helle Bechmann-Pasquinucci, Nicolas J. Cerf, Miloslav Dušek, Norbert Lütkenhaus, and Momtchil Peev. 2009. The Security of Practical Quantum Key Distribution. *Reviews of Modern Physics* 81 (2009), 1301–1350. Issue 3.

[31] Peter W. Shor. 1997. Polynomial-Time Algorithms for Prime Factorization and Discrete Logarithms on a Quantum Computer. *SIAM J. Comput.* 26, 5 (1997), 1484–1509.

[32] Peter W. Shor and John Preskill. 2000. Simple Proof of Security of the BB84 Quantum Key Distribution Protocol. *Physical Review Letters* 85 (2000), 441–444. Issue 2.

[33] Gayane Vardoyan, Saikat Guha, Philippe Nain, and Don Towsley. 2021. On the Stochastic Analysis of a Quantum Entanglement Distribution Switch. *IEEE Transactions on Quantum Engineering* 2 (2021), 1–16. doi:10.1109/tqe.2021.3058058

[34] Lorenza Viola, Emanuel Knill, and Raymond Laflamme. 2001. Constructing Qubits in Physical Systems. *Journal of Physics A: Mathematical and General* 34, 35 (2001), 7067–7079.

A Memory Decoherence and Quantum Bit Error Rates

We now provide a more detailed explanation of the underlying physical process in the quantum router.

The central parameter in quantum key generation is the secret fraction, which is to be optimized. In multiplexing, the additionally introduced quantum memories mainly influence this parameter. During storage, the qubits decohere in the memories, meaning their quantum state changes. Recalling (2), the amount of *decoherence* can be characterized by the *qubit fidelity*, which is a distance measure between two different qubit states (here, the initial and the decohered state):

$$F(\delta) = \exp(-\delta/\tau),$$

where δ is the age of the qubit, and τ is a decoherence constant specific to the quantum memory. The qubit age is an integer $\delta \in \mathbb{N}_0$ measured in entanglement generation attempts (rounds).

Initially, the quantum state of each party k is given by $\rho_k = |\phi^+\rangle\langle\phi^+|$. Due to the time evolution caused by the decoherence, the state changes to the depolarized state $\rho_k^{dep} = \sum_{i=0}^{3} K_i \rho K_i^\dagger$ with the Kraus operators as follows:

$$K_0 = \mathbb{1}_2 \otimes \sqrt{F}\,\mathbb{1}_2$$
$$K_i = \mathbb{1}_2 \otimes \sqrt{\frac{1-F}{3}}\,\sigma_i, \quad i \in \{x, y, z\}.$$

$\sigma_{x,y,z}$ (in the following also denoted as X, Y, Z) are the Pauli matrices acting on a single qubit. This yields

$$\begin{aligned}
&\rho_k^{dep}(F_k)\\
&= \quad F_k\,|\phi^+\rangle\langle\phi^+| + \frac{1-F_k}{3}\left(|\phi^-\rangle\langle\phi^-| + |\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|\right)\\
&= \begin{bmatrix} \frac{F_k}{3}+\frac{1}{6} & 0 & 0 & \frac{2}{3}F_k - \frac{1}{6} \\ 0 & \frac{1}{3}-\frac{F_k}{3} & 0 & 0 \\ 0 & 0 & \frac{1}{3}-\frac{F_k}{3} & 0 \\ \frac{2}{3}F_k-\frac{1}{6} & 0 & 0 & \frac{F_k}{3}+\frac{1}{6} \end{bmatrix}
\end{aligned}$$

with Bell states

$$|\phi^\pm\rangle = 1/\sqrt{2}(|00\rangle \pm |11\rangle) \text{ and } |\psi^\pm\rangle = 1/\sqrt{2}(|01\rangle \pm |10\rangle).$$

The total state of $2N$ qubits on which the entanglement swapping is performed is given by the Tensor product of all bipartite entangled states between the end nodes C and all P_i:

$$\rho_{tot} = \otimes_{k=1}^{N} \rho_k^{dep}(F_k). \tag{10}$$

Given ρ_{tot}, the GHZ measurement is performed: First, one party (here, party C) performs a controlled-NOT (or, CNOT) operation with each P_i, i.e., party P_i flips its qubit if C's qubit is in state $|1\rangle$. In a second step, party C performs a Hadamard operation H on its qubit, thus producing a superposition $(|0\rangle \pm |1\rangle)$ of its own qubit. The final entangled state is calculated via:

$$\rho_{fin} = H \prod_{i=1}^{N-1} CNOT_i \rho_{tot} \left(\prod_{i=1}^{N-1} CNOT_i\right)^T H^T. \tag{11}$$

Finally, a measurement in the X-basis is performed on each qubit. The total entanglement swapping protocol results in a GHZ diagonal state given by its density matrix ρ^{out} of dimension $2^N \times 2^N$ of which each party holds one qubit:

$$\begin{aligned}
\rho^{out} &= \lambda_0^+ |GHZ_0^+\rangle\langle GHZ_0^+| + \lambda_0^- |GHZ_0^-\rangle\langle GHZ_0^-| + \\
&\sum_{\ell=1}^{(2^N-2)/2} \lambda_\ell \left(|GHZ_\ell^+\rangle\langle GHZ_\ell^+| + |GHZ_\ell^-\rangle\langle GHZ_\ell^-|\right).
\end{aligned} \tag{12}$$

The GHZ states are given as $|GHZ_\ell^\pm\rangle = 1/\sqrt{2}\left(|\ell\rangle \pm |N^2 - 1 - \ell\rangle\right)$ with the states written in binary notation. An explicit calculation leads to:

$$\begin{aligned}
\rho_{(1,1)}^{out} = \rho_{(2^N,2^N)}^{out} &= \frac{\lambda_0^+ + \lambda_0^-}{2}\\
&= \chi_{norm} \prod_k \rho_{k,(1,1)}^{dep} + \prod_k \rho_{k,(2,2)}^{dep}\\
&= \chi_{norm} \left(\prod_{k=1}^{N}\left(\frac{F_k}{3}+\frac{1}{6}\right) + \prod_{k=1}^{N}\left(\frac{1}{3}-\frac{F_k}{3}\right)\right),
\end{aligned} \tag{13}$$

$$\begin{aligned}
\rho_{(2^N,1)}^{out} = \rho_{(1,2^N)}^{out} &= \frac{\lambda_0^+ - \lambda_0^-}{2}\\
&= \chi_{norm} \prod_k \rho_{k,(4,1)}^{dep}\\
&= \chi_{norm} \prod_{k=1}^{N}\left(\frac{2}{3}F_k - \frac{1}{6}\right),
\end{aligned} \tag{14}$$

$$\tag{15}$$

and all other entries along the diagonal are given by the λ_ℓ with $\ell \in \{1, (2^N - 2)/2\}$ of the form

$$\chi_{norm}\left(\rho_{C,(1,1)}^{out} \cdot \prod_{k=2}^{N} \rho_{k,(x,x)}^{out} + \rho_{C,(2,2)}^{out} \cdot \prod_{k=2}^{N} \rho_{k,(\bar{x},\bar{x})}^{out}\right)$$

with indices (x, x) either being $(1, 1)$ or $(3, 3)$. The indices $(\bar{x}, \bar{x})$ belong to the negation of (x, x), i.e., they are given by $(2, 2)$ as the negation of $(1, 1)$ or $(4, 4)$ as the negation of $(3, 3)$, respectively. Along the first half of the diagonal, the entries are given by all combinations of indices. They are ordered following the binary representation of the belonging index ℓ from the corresponding λ_ℓ with 0 representing the indices $(1, 1)$ (respectively, $(2, 2)$ for the negation) and 1 representing the indices $(3, 3)$ (respectively, $(4, 4)$ for the negation). Due to symmetry, we find the same entries in the second half of the diagonal in reverse order, see Eq. (12). The normalization constant is given by $\chi_{norm} = 2^{N-1}$ so that $Tr(\rho^{out} = 1)$ holds.

As a concrete example we give the resulting density matrix for the tripartite network:

$$
\begin{aligned}
\rho^{out}_{(1,1)} &= \frac{\lambda_0^+ + \lambda_0^-}{2} = 2^2 \left(\rho^{dep}_{C,11}\rho^{dep}_{P_2,11}\rho^{dep}_{P_2,11} + \rho^{dep}_{C,22}\rho^{dep}_{P_2,22}\rho^{dep}_{P_2,22}\right)\\
&= \rho^{out}_{(8,8)},\\
\rho^{out}_{(2,2)} &= \lambda_1 = 2^2 \left(\rho^{dep}_{C,11}\rho^{dep}_{P_2,11}\rho^{dep}_{P_2,33} + \rho^{dep}_{C,22}\rho^{dep}_{P_2,22}\rho^{dep}_{P_2,44}\right)\\
&= \rho^{out}_{(7,7)},\\
\rho^{out}_{(3,3)} &= \lambda_2 = 2^2 \left(\rho^{dep}_{C,11}\rho^{dep}_{P_2,33}\rho^{dep}_{P_2,11} + \rho^{dep}_{C,22}\rho^{dep}_{P_2,44}\rho^{dep}_{P_2,22}\right)\\
&= \rho^{out}_{(6,6)},\\
\rho^{out}_{(4,4)} &= \lambda_3 = 2^2 \left(\rho^{dep}_{C,11}\rho^{dep}_{P_2,33}\rho^{dep}_{P_2,33} + \rho^{dep}_{C,22}\rho^{dep}_{P_2,44}\rho^{dep}_{P_2,44}\right)\\
&= \rho^{out}_{(5,5)}, \text{ and}\\
\rho^{out}_{(8,1)} &= \frac{\lambda_0^+ - \lambda_0^-}{2} = 2^2 \left(\rho^{dep}_{C,41}\rho^{dep}_{P_2,41}\rho^{dep}_{P_2,41}\right) = \rho^{out}_{(1,8)}.
\end{aligned}
$$

Note that the equalities $\rho^{out}_{(1,1)} = \rho^{out}_{(2^N,2^N)}$ and $\rho^{out}_{(1,2^N)} = \rho^{out}_{(2^N,1)}$ hold due to the equality of the corresponding terms in ρ^{dep}_k. Due to the GHZ measurement performed, the output state is projected onto one of the GHZ states given above. Depending on the measurement output, the parties can perform local operations on their qubit to turn the entangled state into the $|GHZ_0^+\rangle$ state.

This state can then be used to distribute a secret key between all parties. The whole multipartite BB84 protocol is given in [24]. The key is generated by measuring the own qubit in the X-basis and recording the classical measurement outcome. Since the state is a genuine multipartite entangled state, each party should ideally get the same measurement outcome and, therefore, an identical bit string for the raw key. The additional information from the Z-basis measurements is used for some test rounds. Since the input states already have reduced fidelities depending on the qubit ages, the output state also has reduced fidelity. This relates to the *quantum bit error rates* (QBERs). These QBERs are calculated based on the classical information from the measurements. This is done by partially comparing the measured outcome. The QBER gives the ratio between different measurement results and the total number of measurement outcomes. For the multipartite setup, measurements in the X-basis and the Z-basis are considered separately:

$$e_X = \frac{1 - \langle X^{\otimes N}\rangle}{2}, \tag{16}$$

$$e_{CP_i} = \frac{1 - \langle Z_C Z_{P_i}\rangle}{2} = Prob(Z_C \neq Z_{P_i}). \tag{17}$$

The error rate in the X-basis is described by e_X, and e_{CP_i} gives the bipartite error rates between party C and each party P_i, respectively. With $X = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}$, it holds $\langle X^{\otimes N}\rangle = Tr\left(\rho^{out} X^{\otimes N}\right) = (\lambda_0^+ - \lambda_0^-)$. Explicitly, we find for the X-error:

$$
\begin{aligned}
e_X &= \frac{1}{2} - \chi_{norm} \prod_{i=1}^{N} \left(\frac{2}{3}F_k - \frac{1}{6}\right)\\
&= \frac{1}{2} - \frac{1}{2 \cdot 3^N} \prod_{1=1}^{N} (4F_k - 1)\,.
\end{aligned} \tag{18}
$$

For the bipartite error rate, the probability that $Z_C \neq Z_{P_i}$ holds is given for all matrix entries containing either $\rho^{dep}_{C,(1,1)}\rho^{dep}_{P_i,(2,2)}$ or $\rho^{dep}_{C,(2,2)}\rho^{dep}_{P_i,(1,1)}$. The bipartite error rate e_{CP_i} only depends on the fidelities of party C and that specific P_i. Explicitly, all other terms with fidelities $F_{P_{i'}}$ with $P_{i'} \neq P_i$ cancel out, leading to a constant prefactor $(1/2)^{N-2}$. Explicitly, we have

$$
\begin{aligned}
e_{CP_i} &= Prob(Z_C \neq Z_{P_i})\\
&= 2 \cdot \chi_{norm} \cdot \left(\frac{1}{2}\right)^{N-2} \cdot\\
&\quad \left[\left(\frac{F_C}{3} + \frac{1}{6}\right)\left(\frac{1}{3} - \frac{F_{P_i}}{3}\right) + \left(\frac{1}{3} - \frac{F_C}{3}\right)\left(\frac{F_{P_i}}{3} + \frac{1}{6}\right)\right]\\
&= \frac{2}{9}\left(F_C + F_{P_i} - 4F_C F_{P_i} + 2\right).
\end{aligned} \tag{19}
$$

Due to these errors, the ratio of secret bits that can be generated in the BB84 protocol is diminished. This figure of merit is described by the *secret fraction* r_∞:

$$r_\infty\left(\bar{e}_X, \bar{e}_{CP_i}\right) = \max\left\{0, 1 - h\left(\bar{e}_X\right) - \max_i h\left(\bar{e}_{CP_i}\right)\right\}, \tag{20}$$

where $\bar{e}_X$ and $\bar{e}_{CP_i}$ are the QBERs averaged over one matching and h is the binary Shannon entropy $h(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. The secret key rate gives the fraction of secret bits from all shared bits (given by the router rate R) from which a secret key can be generated:

$$K = r_\infty\left(\bar{e}_X, \bar{e}_{AB_i}\right) R. \tag{21}$$

The router rate gives the number of GHZ measurements l that are established on average per round t and per memory per party:

$$R = \frac{1}{m}\sum_{t=1}^{t_C} l(t), \tag{22}$$

where m is the number of memories per party and the sum is taken over all rounds t up to a current round t_C. Note that the number of GHZ measurements performed in each round is given by the cardinality of the underlying router matching considered here, i.e., $l = |M|$.

Paper B

Multipartite multiplexing strategies for quantum routers

Julia A. Kunzelmann, Hermann Kampermann, and Dagmar Bruß
Institute for Theoretical Physics III, Heinrich Heine University Düsseldorf, 40225 Düsseldorf, Germany

(Received 1 July 2024; accepted 3 September 2024; published 18 September 2024)

This work explores the important role of quantum routers in communication networks and investigates the increase in efficiency using memories and multiplexing strategies. Motivated by the bipartite setup introduced by Abruzzo *et al.* [Phys. Rev. A **89**, 012303 (2014)] for finite-range multiplexing in quantum repeaters, we extend the study to an N-partite network with a router as a central station. We present a general protocol for N parties after defining the underlying matching problem and we calculate the router rate for different N. We analyze the improvement due to multiplexing and analyze the secret key rate with explicit results for the tripartite network. Investigating strategic qubit selection for the Greenberger-Horne-Zeilinger measurements, we show that using cutoffs to remove qubits after a certain number of rounds and consistently combining qubits with the lowest number of storage rounds leads to an optimal secret key rate.

DOI: 10.1103/PhysRevA.110.032617

I. INTRODUCTION

Quantum communication is a major field of research in quantum information theory. An essential area is the generation of secret keys, which can be used in cryptography. In quantum key distribution (QKD), such keys are generated between two parties [1,2]. The generalization to N parties is called conference key agreement (CKA) [3,4]. Usually, photons are used to distribute a key which limits the distance to about 150 km [5] as photon losses scale exponentially with distance. Quantum repeaters are needed to overcome this problem [6,7]. In an intermediate repeater station, the entangled state between the remote parties is established by performing a Bell state measurement (BSM). So far, quantum repeaters connecting two parties have been investigated, either without memories [8] or with memories [9]. In the latter case, additional multiplexing can be used to perform parallel independent Bell state measurements, thus increasing the generation rate of entangled states between the parties [10,11] per round. These states can, e.g., be used for quantum key distribution such as the BB84 protocol [1] or (measurement) device-independent QKD protocols [2,12].

In this work we introduce a generalization to a quantum router that connects N parties in a star graph, where the quantum router is the central node. This central station is used to distribute multipartite entanglement between the parties in larger networks [13]. Similar network structures have been analyzed in quantum switches with or without buffer, where the goal is to connect $N \leqslant k$ of the k users via an entangling measurement. This setup has been investigated numerically [14] and also analytically using Markov chains [15]. Here we deal with the distribution of entangled Greenberger-Horne-Zeilinger (GHZ) states between N parties by performing GHZ measurements within the router. We additionally include quantum memories for multiplexing in the quantum router by generalizing the protocol from [11]. We define the underlying N-dimensional matching problem, discuss suitable algorithms for different network sizes, and analyze the router rate. In a second step, we analyze such networks in the context of conference key agreement. Our setup can be seen as a generalization of the measurement-device-independent QKD protocol with quantum memories from [9] to more than two parties. Compared to previous work about CKA in star graphs [16], we additionally analyze the effect of quantum memories and the use of multiplexing. We calculate the quantum bit error rates for the BB84 protocol with N parties and use this to determine the asymptotic secret fraction and the secret key rate. The focus is on examining different matching strategies to select the qubits for the GHZ measurements to maximize the key rate.

The paper is structured as follows. In Sec. II we present the N-partite network with the quantum router as the central element and explain the entanglement distribution among all parties. In Sec. III we introduce multipartite multiplexing and define the related matching problem from graph theory. We further focus on the router rate, i.e., the rate with which entangled states can be created in each round of the protocol. Router rates for different setups are calculated. Finally, we consider conference key agreement and determine secret key rates for tripartite networks in Sec. IV. We further analyze different strategies for minimizing the influence of memory decoherence for the tripartite network. We conclude in Sec. V with a summary and outlook.

II. QUANTUM ROUTER WITH MEMORIES IN N-PARTITE STAR GRAPHS

We first generalize the concept of a quantum repeater to a quantum router in a network of N communicating parties that are located at equal distances around the router. The scenario with unequal distances of the parties could be analyzed in an analogous way. The general setup of such a star-shaped network is shown in Fig. 1. The entanglement distribution is performed in the following steps.

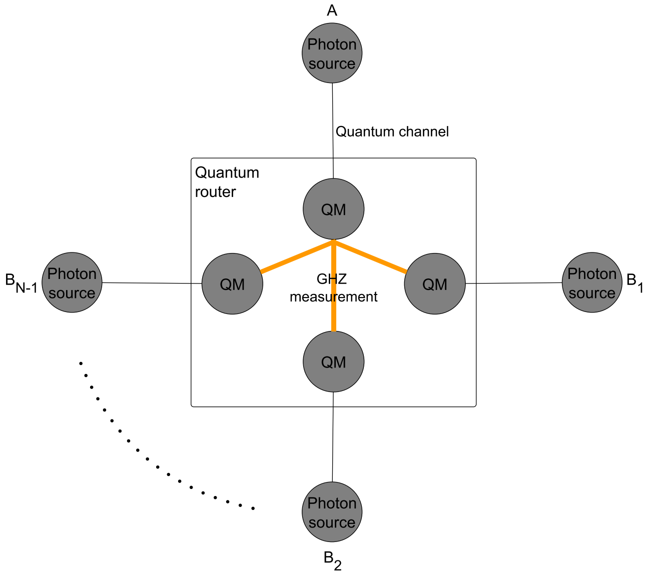


FIG. 1. General setup of a quantum router with quantum memories (QM) that is connected to N parties. All parties are placed at equal distances around the router. The photon sources of the parties each produce a Bell state, of which one qubit is held locally while the second qubit is sent to the central station. A GHZ measurement between all parties is performed.

(i) Each party prepares a Bell state $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ and sends one qubit via the quantum channel to the quantum router. The second qubit is held locally by each party. Qubits that successfully arrive at the router are stored in a memory.

(ii) In each round, in which some memories of all parties are filled, a GHZ measurement is performed (Fig. 2), where one party (here party A) has a special role, providing the control qubit for the controlled-NOT gates and performing the Hadamard gate. The measurement outcome is announced to all parties. Memories, whose stored qubits are included in a GHZ measurement, are reset for the next round. Filled memories, which are not included in a GHZ measurement, remain filled for the next round.

(iii) Depending on the measurement outcome, the parties perform a phase flip (party A) or bit flip (parties B_i), if necessary, in order to obtain the desired GHZ state $|\text{GHZ}\rangle_N = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes N} + |1\rangle^{\otimes N})$.

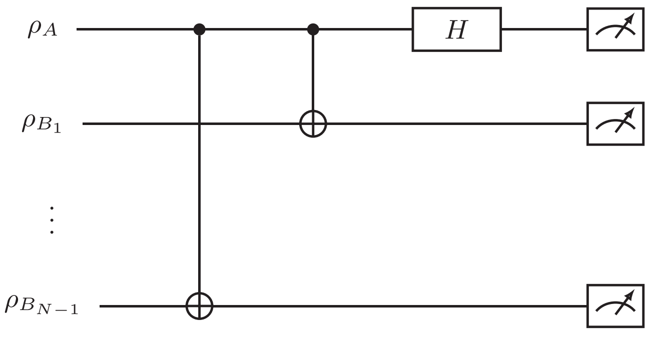


FIG. 2. Quantum circuit for performing the entangling GHZ measurement. Gates are only applied to the qubits stored in the memories.

We call these three steps together one round. In the following, the label of the round is denoted by s. Any number of rounds can be performed, each containing one attempt of entanglement generation. By including $m > 1$ memories per party, the probability that each party has at least one filled quantum memory in a round can be increased. This means that the number of distributed states in a single round increases to a maximum of m. This so-called multiplexing is examined in detail in Sec. III.

As a figure of merit, we define the router rate as the number of successful GHZ measurements per memory per round averaged over the whole running time up to a current round s_c, namely,

$$R(s_c) = \frac{1}{s_c}\sum_{s=1}^{s_c} \frac{\langle l\rangle(s)}{m}. \tag{1}$$

Here $\langle l\rangle(s)$ is the average number of GHZ measurements in round s. During the storage process, the qubits are subject to noise, which is analyzed in the following.

Noise model for the memories

We model noise affecting the stored qubits by depolarization. Starting the storage in round s_0 with a given quantum state ρ_0, the depolarized state in round s has the form

$$\rho(s - s_0) = p(s - s_0)\rho_0 + \frac{1 - p(s - s_0)}{2}\mathbb{1}. \tag{2}$$

The decoherence parameter τ of the memory is related to the probability of white noise [see Eq. (2)] by

$$p(\delta) = e^{-\delta/\tau}, \tag{3}$$

with $\delta = s - s_0$ the number of storage rounds of a qubit. Note that the decoherence parameter τ and the number of storage rounds δ are each given by an integer.

The bipartite states after decohering in the memories are given by

$$\begin{aligned}\rho_i^{dep} =& F_i|\phi^+\rangle\langle\phi^+| + \frac{1 - F_i}{3}(|\phi^-\rangle\langle\phi^-| \\ &+ |\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|),\end{aligned} \tag{4}$$

with $F_i = \frac{1}{4} + \frac{3}{4}p(\delta)$ defining the fidelity of the states for the parties $i \in \{A, B_1, B_2, \dots\}$. The total input state $\rho^{\text{dep}}_{AB_1\cdots B_{N-1}}$ is given by the tensor product of the noisy states provided by each party. Performing the GHZ measurement on those qubits which are stored in the memories, the parties end up sharing a GHZ diagonal state $\tilde{\rho}^{\text{dep}}_{AB_1\cdots B_{N-1}}$ of the remaining qubits held locally. It is related to the input fidelities F_i of the initial depolarized states by its GHZ diagonal elements. The fidelity of the output state is given by $\tilde{F}_{\tilde{\rho}^{\text{dep}}} = \langle\text{GHZ}|\tilde{\rho}^{\text{dep}}_{AB_1\cdots B_{N-1}}|\text{GHZ}\rangle_N$. An explicit calculation for the tripartite case is given in Appendix C.

III. *N*-PARTITE MULTIPLEXING

The generalized setup of a quantum router with multiplexing is shown in Fig. 3. All parties have a fixed number m

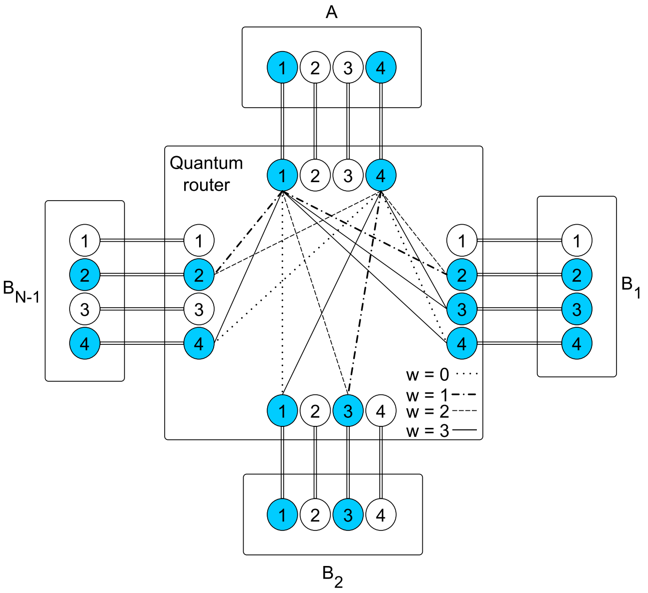


FIG. 3. Generalized setup of a quantum router with multiplexing for an N-partite star graph. Shown is an exemplary filling of the memories. Empty memories are white, while filled memories are shown in blue. The connection lengths w (difference of labels of filled memories) for this example are also indicated.

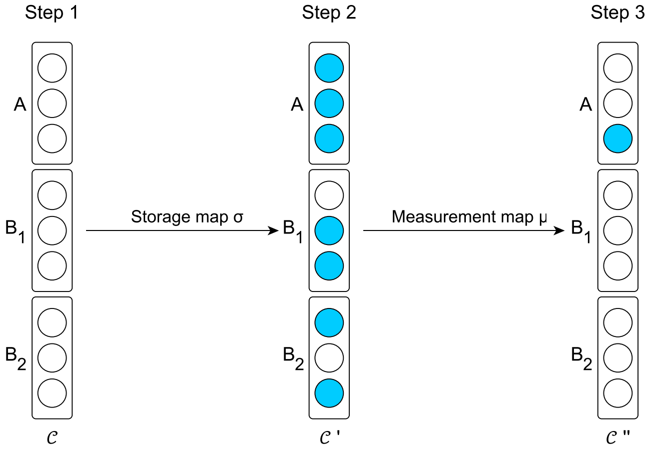


FIG. 4. Temporal structure of a round in the multiplexing protocol. In each round, qubits are first sent to the router and if they arrive successfully are stored in the respective memories (storage map σ leading from configuration $\mathcal{C}$ to $\mathcal{C}'$). The GHZ measurements are performed and memories that are involved in a measurement are emptied (measurement map μ). The new memory configuration $\mathcal{C}''$ forms the starting configuration $\mathcal{C}$ for the following round.

of photon sources and correspondingly m quantum memories in the router. The memories within the quantum router are either empty (white in Fig. 3) or filled with a qubit (blue in Fig. 3). The number of filled memories depends on the loss rate η of the channel and the probability of successful storage. The parameter η relates to the distance d between each party and the central router via $d = -10/\alpha \log_{10} \eta$, with the fiber attenuation coefficient α. In the following, we also call η the transmittivity of the quantum channel. The connection length w is defined as the distance between the memories, i.e., the difference of the labels (see the example in Fig. 3). Physically, this parameter is relevant in an experimental setup where it is not possible to connect quantum memories with larger physical distances. For each memory configuration in every round, the protocol aims to perform the maximal number of GHZ measurements between all parties.

A. Time structure of a multiplexing protocol round

The different configurations of the memories the quantum router goes through in one round of the entanglement distribution protocol are shown in Fig. 4. Each memory configuration is given by a vector $\mathcal{C}$ (for details of its notation see Sec. III C), where each entry represents one memory within the router. Following the steps for entanglement distribution given in Sec. II, one finds two transitions of the memory configurations: Starting a round with a memory configuration $\mathcal{C}$, it changes to the intermediate configuration $\mathcal{C}'$ after sending and storing one part of the Bell pairs prepared by the parties. The qubit arrives with probability η and is then heralded and stored by the quantum router. Based on the configuration $\mathcal{C}'$, a maximal number of GHZ measurements l is performed. In each round, the router reports whether the measurement was successful. If so, the measurement results and the information on which memories were involved in the measurement is communicated. The final memory configuration after resetting the used memories is given by $\mathcal{C}''$. This memory configuration is kept for the next round.

In the following, the focus will be on the choice of the memories included in a GHZ measurement. Finding a combination that maximizes the number of GHZ measurements l per round corresponds to the problem of matching from graph theory, which is introduced in the following section.

B. Matching problem

The memories of the quantum router form an N-partite graph $G = (V, E)$ consisting of nodes V (here the filled memories) and edges E (the connectability between the memories). The set of nodes V is divided into N pairwise disjoint subsets $V_1, V_2, \dots, V_N$ which result from the allocation of the memories to the different parties. An edge always connects two nodes from different subsets, i.e., $E = \{\{v_i, v_j\} \mid v_i \in V_i \wedge v_j \in V_j \text{ for } i \neq j\}$. Since the goal of the router is to perform a GHZ measurement between party A and each B_i, we define hyperedges (sets of nodes) which need to fulfill the following properties.

(i) Each hyperedge always consists of a set of N nodes from different subsets, i.e., $T = \{\{v_1, \dots, v_N\} \mid v_1 \in V_1, \dots, v_n \in V_N\} \subseteq V_1 \times \dots \times V_N$, containing exactly one memory per party.

(ii) The vertices are connected in such a way that each B_i has exactly one edge to party A but no edges to other B_i or the own subset since this is fixed by the GHZ measurement circuit (see Fig. 2). In general, nodes can appear in several hyperedges. Party A can be fixed for the whole protocol or it can be chosen individually in each round.

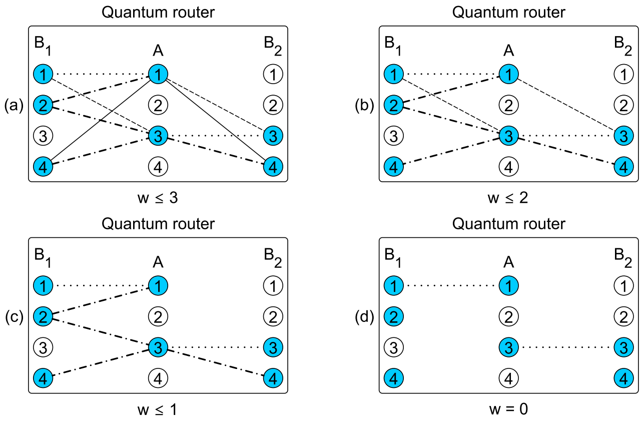


FIG. 5. Example of a tripartite graph in the quantum router for different choices of the maximal connection length w. (a) Graph for full-range multiplexing such that any node of one subset can be connected to any node of another subset. In (b) and (c) the maximal connection length is reduced to 2 and 1, respectively. (d) Finally, only parallel connections are allowed ($w = 0$) so that some filled memories are no longer considered in the matching. Solid lines represent connections with $w = m - 1 = 3$ (maximal), dashed lines represent $w = 2$, dash-dotted lines represent $w = 1$, and dotted lines show connections belonging to $w = 0$.

(iii) In the case of restricting the connection length w, edges are only allowed to be drawn between two nodes fulfilling that constraint.

We point out that finding a set of such hyperedges in a given graph corresponds to the modified matching problem we consider here. Its formal definition is given below. In every protocol round the graph in the quantum router is constructed based on the given memory configuration. Memories only contribute to the graph if they are filled. Hyperedges are drawn between the memories such that they fulfill the properties defined above. The graph construction for a fixed memory configuration but different connection lengths w is exemplarily shown in Fig. 5. For example, we find the following hyperedges for the graph in Fig. 5(c): $\{\{2,3,3\},\{2,3,4\},\{4,3,3\},\{4,3,4\}\}$. Here we fix party A for the whole protocol, since the increased rate due to a dynamic choice of a different Alice in each round is small compared to increasing the connection length. Additionally, it is not clear whether this can be realized easily in an experimental setup.

Since the goal of the multiplexing scheme is to perform a maximal number of GHZ measurements per round, we want to find a set of pairwise disjoint hyperedges where the set has maximal cardinality. The condition of being pairwise disjoint follows from the fact that a stored qubit cannot be used in two different GHZ measurements. A set of hyperedges in which no two hyperedges share a common node is called a matching. A set of maximum cardinality concerning the number of contained hyperedges is called maximum matching. Note that more than one maximum matching may exist. In the previous example of Fig. 5(c), each allowed hyperedge is a valid matching with a maximum cardinality of 1. A larger set of hyperedges cannot be formed, since all hyperedges share Alice's third node as a common node. Figure 6 shows an explicit example of a valid maximum matching given each graph from Fig. 5 as an input instance.

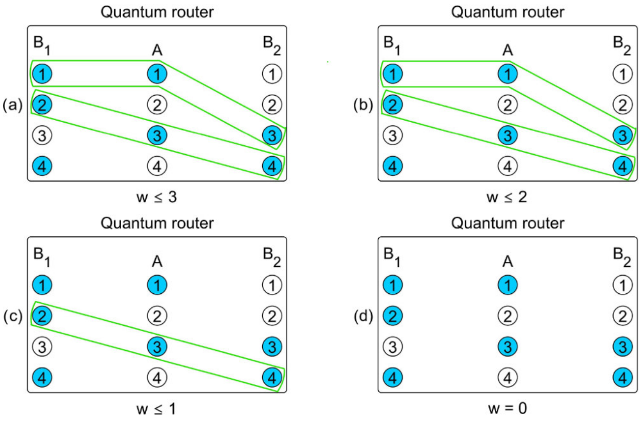


FIG. 6. Examples of maximum matching from the corresponding graphs in Fig. 5 for different connection lengths. For (a) $w \leqslant 3$ and (b) $w \leqslant 2$, there is a maximum matching with two hyperedges each, and for (c) $w \leqslant 1$, there is a maximum matching with one hyperedge. For (d) $w = 0$, no matching is found. All hyperedges are shown in green. Note that this choice of hyperedges is not unique.

The underlying problem from graph theory is formally defined as follows:

Maximum N-dimensional matching	
Given:	An N-partite graph instance and all valid hyperedges $T \subseteq V_1 \times V_2 \times \cdots \times V_N$
Question:	What is a valid maximum matching $M \subseteq T$?

However, note that in our scenario, not all possible N-partite graph instances can occur due to the given physical constraints. The maximum number of hyperedges (defining the number of GHZ measurements l that can be performed in each round) is limited by the given graph instance in every round, namely, by the number of existing edges between the nodes [given by the degree $\deg(v_j)$ of each node v_j], and by the number of filled memories n_k per party k:

$$l \leqslant \min_{k\in\{A,B_i\}} \{|\{v_j|\deg(v_j) > 0 \wedge v_j \in V_k\}|\} \leqslant \min_{k\in\{A,B_i\}} \{n_k\}. \quad (5)$$

Recall that the degree of a node may depend on the maximal connection length w. For full-range multiplexing it holds that $l = \min_{k\in\{A,B_i\}}\{n_k\}$.

Finding a matching in this case as well as for $w = 0$ is trivial. The original maximum N-dimensional matching problem is contained in the class $\mathcal{NP}$ [17,18]. Nevertheless, it is possible to construct algorithms that solve the previously introduced modified maximum N-dimensional matching problem for small graphs if the total number of memories does not become too large. In that case, it is still possible to go through all hyperedges and find a combination of pairwise disjoint hyperedges that leads to maximum cardinality. For $N = 3$ parties, the problem can also be solved via network flow. For larger N, all combinations of hyperedges are to be considered and the optimal one is chosen. Details about the implementation can be found in Appendix B.

In addition to finding a maximum matching, weights can be added to the edges and optimized in a second step. In our scenario, the weights will be a function of the number of rounds that the relevant qubits have spent in the memories. Note that the main goal does not change and a maximum matching is to be found first. If there is more than one such maximum matching, the one with optimized weights is chosen. This leads to the maximum N-dimensional matching with weights. Here it is always necessary to go through all solutions and choose the optimal matching. How the weights are set is defined in Sec. IV A. The implementation details are given in Appendix B.

C. Router rate

In this section we analyze the router rate

$$R(s_c) = \frac{1}{s_c}\sum_{s=1}^{s_c}\frac{\langle l\rangle(s)}{m}, \tag{6}$$

as defined in Sec. II. Following [11], we find analytical expressions for the generalized rate in a quantum router connecting more than two parties. We start by defining the configuration of the memories belonging to each party via the vectors $\vec{a}, \vec{b}_1, \vec{b}_2, \ldots$, each of length m. Every entry in a vector represents one memory from the set of memories per party and can be either 0 (empty memory) or 1 (filled memory). In the example given in Fig. 5, the vectors describing the memory configurations are $\vec{a} = (1, 0, 1, 0)$, $\vec{b}_1 = (1, 1, 0, 1)$, and $\vec{b}_2 = (0, 0, 1, 1)$. From this, the total memory configuration $\mathcal{C} = (\vec{a}_1, \vec{b}_1, \ldots, \vec{b}_{N-1})$ follows, a vector that is given by the concatenation of the memory configuration of each party. Again, each entry $c_i \in \{0, 1\}$ in $\mathcal{C}$ represents one memory. The value is set to 0 for empty memories and it is set to 1 when the memory is filled. As the maximal connection length w limits the number of GHZ measurements, the set of configurations of m memories per party leading to l GHZ measurements is further denoted by $\mathcal{H}_w^m(l)$. Regarding Fig. 5 as an example, then $(\vec{a}, \vec{b}_1, \vec{b}_2) \in \mathcal{H}_3^4(2)$, $(\vec{a}, \vec{b}_1, \vec{b}_2) \in \mathcal{H}_2^4(2)$, $(\vec{a}, \vec{b}_1, \vec{b}_2) \in \mathcal{H}_1^4(1)$, and $(\vec{a}, \vec{b}_1, \vec{b}_2) \in \mathcal{H}_0^4(0)$, resulting in no matching. However, this configuration does not allow, e.g., $l = 2$ and $w = 1$, i.e., $(\vec{a}, \vec{b}_1, \vec{b}_2) \notin \mathcal{H}_1^4(2)$,

With this notation, the transition between two memory configurations can be denoted by the storage map $\sigma_l : \mathcal{H}_w^m(0) \to \mathcal{H}_w^m(l)$ and the measurement map $\mu_l : \mathcal{H}_w^m(l) \to \mathcal{H}_w^m(0)$ (see also Fig. 4). The former contains the transition from configuration $\mathcal{C}$ to $\mathcal{C}'$ given by the probability η of successfully sending each of the qubits through the quantum channel:

$$\begin{aligned}\mathrm{Prob}[\sigma_l(\mathcal{C}) = \mathcal{C}'] &= \mathrm{Prob}(\mathcal{C}'|\mathcal{C})\\ &= \prod_{i=1}^{N\cdot m}\mathrm{Prob}(c_i'|c_i)\\ &= \prod_{i=1}^{m}\mathrm{Prob}(a_i'|a_i)\prod_{j=1}^{N-1}\mathrm{Prob}(b_{j,i}'|b_{j,i}).\end{aligned} \tag{7}$$

The transition probability between an initial configuration $\mathcal{C}$ and the configuration $\mathcal{C}'$ is calculated memorywise for each configuration entry c_i and c_i' with $c_i, c_i' \in \{0, 1\}$ and $i \in \{1, 2, \ldots, Nm\}$ (i.e., a_i for Alice's memories and $b_{j,i}$ with $i \in \{1, \ldots, m\}$ for the memories of the j Bobs) by

$$\mathrm{Prob}(c_i'|c_i) = (1-\eta)(1-c_i')(1-c_i) + \eta c_i'(1-c_i) + c_i'c_i. \tag{8}$$

The map μ_l describes transitions given by the GHZ measurements, i.e., it maps the memory configurations $\mathcal{C}'$ before the measurements to after the measurements ($\mathcal{C}''$). The choice of the memories included in such a measurement is made based on the underlying matching problem explained in the preceding section. Combining both maps, the probability for an initial memory configuration $\mathcal{C}$ to end in the final configuration $\mathcal{C}''$ is given by

$$\mathrm{Prob}[\mu_l \circ \sigma_l(\mathcal{C}) = \mathcal{C}''] = \sum_{\mathcal{C}' \in \mathcal{H}_w^m(l)} \delta_{\mu_l(\mathcal{C}'),\mathcal{C}''}\mathrm{Prob}[\sigma_l(\mathcal{C}) = \mathcal{C}'], \tag{9}$$

where δ denotes the Kronecker delta, which is 1 if and only if $\mu_l(\mathcal{C}') = \mathcal{C}''$ and 0 otherwise.

Remark. The main difference in the calculation of the router rate for $N > 2$ parties (compared to $N = 2$) lies in the determination of these two transition maps. The representation of a configuration $\mathcal{C}$ now includes the concatenation of N vectors, instead of only two vectors. The verification of whether a configuration is in $\mathcal{H}_w^m(l)$ must be adjusted accordingly.

Using these generalized transition matrices and given that all memories are empty at the beginning of the protocol, the router rate can be calculated analytically, analogously to [11]. That is done by computing each possible configuration $\mathcal{C}_s''$ at the end of one round s (which equals the configuration $\mathcal{C}_{s+1}$ at the beginning of the next round) iteratively by knowing the transition probabilities given by Eq. (9) and the final configuration from the end of the previous round denoted by $\mathcal{C}_{s-1}'' = \mathcal{C}_s$:

$$\begin{aligned}\mathrm{Prob}(\mathcal{C}_s'')(s) = \sum_{\mathcal{C}_s \in \mathcal{H}_w^m(0)}\sum_{l=0}^{m}&\mathrm{Prob}[\mu_l \circ \sigma_l(\mathcal{C}_s)\\ &= \mathcal{C}_s'']\mathrm{Prob}(\mathcal{C}_s)(s) = \mathrm{Prob}(\mathcal{C}_{s+1}).\end{aligned} \tag{10}$$

Given this probability for any configuration at the beginning of a round $[\mathrm{Prob}(\mathcal{C})(s)]$, we calculate the probability of having l GHZ measurements:

$$\begin{aligned}&\mathrm{Prob}(\Lambda = l)(s)\\ &\quad= \sum_{\mathcal{C}' \in \mathcal{H}_w^m(l)}\mathrm{Prob}(\mathcal{C}')(s)\\ &\quad= \sum_{\mathcal{C}' \in \mathcal{H}_w^m(l)}\sum_{\mathcal{C} \in \mathcal{H}_w^m(0)}\mathrm{Prob}[\sigma_l(\mathcal{C}) = \mathcal{C}']\mathrm{Prob}(\mathcal{C})(s).\end{aligned} \tag{11}$$

Here Λ denotes a random variable that can take values $0, 1, \ldots, m$ representing the number of performed GHZ measurements. In the case of considering not only deterministic GHZ measurements but also probabilistic measurements, the probability that l GHZ measurements are performed successfully is given by

$$\mathrm{Prob}(\Sigma = l)(s) = \sum_{i=l}^{m} \binom{i}{l} \mathrm{Prob}(\Lambda = i)(s) P_{\mathrm{GHZ}}^{l} (1 - P_{\mathrm{GHZ}})^{i-l}, \tag{12}$$

with P_{GHZ} the success probability of a GHZ measurement. In addition, Σ is a random variable taking values $0, 1, \dots, m$ defining the number of successful GHZ measurements. Then the average number of successful measurements is given by

$$\langle l \rangle(s) = \sum_{l=0}^{m} l \, \mathrm{Prob}(\Sigma = l)(s). \tag{13}$$

Finally, by inserting this in Eq. (6), the router rate can be calculated.

Note that the two matrices describing the transitions of the memory configurations are of dimension $2^{m_{\mathrm{tot}}} \times 2^{m_{\mathrm{tot}}}$, with $m_{\mathrm{tot}} = Nm$ the total number of memories. This limits the network size, for which the router rate can be calculated analytically. For $m_{\mathrm{tot}} > 12$, the runtime becomes infeasible and makes analytical calculations impracticable. All results considering larger networks are based on numerical simulations performing the presented protocol and extracting the average number of GHZ measurements per round from 50 000 repetitions. The simulation is further described in Appendix A.

The router rate for different network sizes is calculated for the following parameters. The probability of the successful transmission of a qubit is $\eta = 0.1$, which corresponds to a distance of $d = 50$ km between the parties and the central node. The fiber coefficient is chosen as $\alpha = 0.2$. This value describes a commercial optical fiber used with light at a wavelength of 1550 nm. The GHZ measurement, as well as the storage process of the qubits in the memories, is assumed to be perfect. In addition, it is assumed that all memories are initially empty and qubits remain in the memories until they are selected for a GHZ measurement by matching. The impact of premature removal of qubits from storage will be discussed later. Figure 7 shows the router rates for a four-partite network with various numbers of memories per party. For each choice of m, maximal connection lengths of $w = 0$, 1, and $m - 1$ are chosen.

In the case of $w = 1$, only a small increase in the router rate with increasing m can be seen in Fig. 7. The biggest difference in the router rate for increasing m is achieved with full-range multiplexing. The plot shows that for small m a large advantage can already be gained by using finite-range multiplexing, e.g., $w = 1$. As the number of memories per party increases, the increase of the router rate with the connection length w becomes larger. In both figures, the difference between $w = 0$ and $w = 1$ for a small number of protocol rounds can be explained by combinatorial arguments. The initial router rate for a connection length $w = 0$ comes from the probability $R(s_c = 1) = \eta^N$ that all memories with the same label are filled after one round [here we find $R(s_c = 1) = 10^{-4}$ for $\eta = 0.1$ and $N = 4$]. For a connection length $w > 0$, all possible combinations of hyperedges that are allowed have to be counted. The probability of ending up in one of these memory configurations normalized over m gives the initial router rate for the multiplexing setup [for $w = 1$, we find 37 memory configurations with one filled memory per party, leading to $R(s_c = 1) = 1.2 \times 10^{-3}$]. Since all memories are initially empty, the difference between the curves for different connection lengths is larger for a small number of rounds. The asymptotic convergence behavior of the curve can be explained by the description of the setup via Markov chains [19,20].

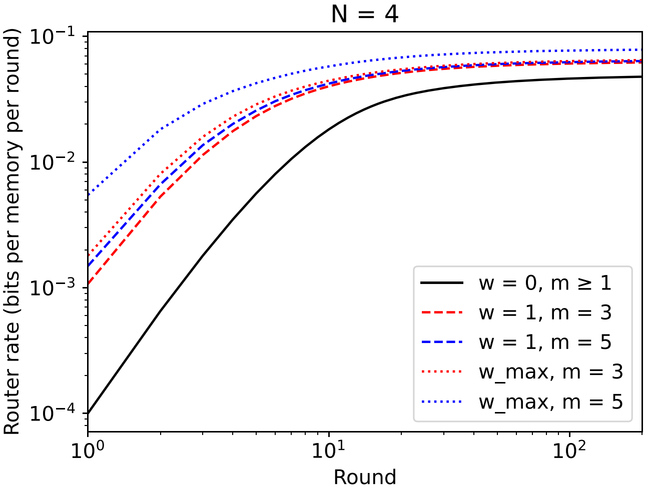


FIG. 7. Comparison of the router rate for different four-partite networks with varying numbers of memories m per party and different maximal connection lengths w. The transmittivity is chosen to be $\eta = 0.1$.

The relationship between the number of communicating parties and the router rate is shown in Fig. 8 for $N = 3$ and 5, with a fixed memory number of $m = 3$. Again, a significant improvement in the router rate can already be achieved by considering finite-range multiplexing with $w = 1$. The simulations also show that for larger N the router rate decreases significantly, but the general behavior of each graph does not change. The fast decrease of the router rate for larger N indicates the importance of the usage of memories in a

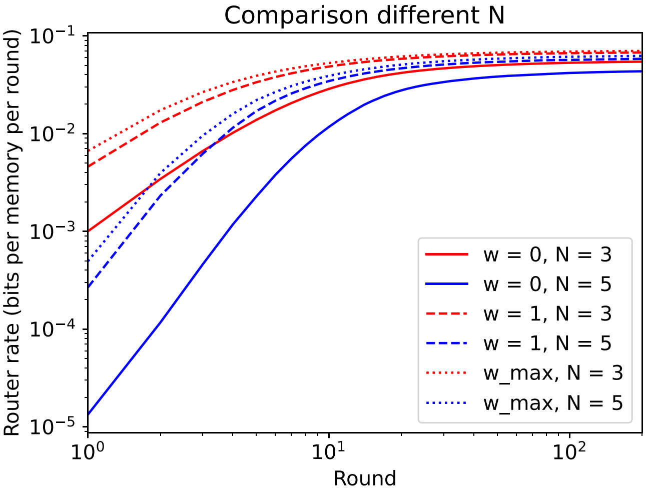


FIG. 8. Comparison of the router rate for different network sizes where each party has $m = 3$ memories. The other parameters are $w = 0, 1, 2$ and $\eta = 0.1$.

quantum router. The effect of decoherence that comes along with the memories is analyzed in the following.

IV. SECRET KEY RATE

When the entangled states are distributed between all parties, these states can be used for further applications such as conference key agreement. This task aims to generate and distribute a common secret key between more than two parties. Given a shared quantum state between N parties, a secure key can be established through local operations and public communication. We use the N-BB84 protocol, a generalization from the bipartite BB84 protocol used for quantum key distribution (for more details, see [3,4]). Given that a GHZ state is shared between all parties, every party measures their local part of the GHZ state either in the Z basis (key generation round) or in the X basis (test round). Additionally, the usual postprocessing steps for CKA are performed as given in [3]. Using some of the measurement outcomes for the classical parameter estimation, the quantum bit error rates (QBERs) Q_X and Q_{AB_i} can be estimated. In particular, Q_{AB_i} is the probability that A and B_i have different outcomes in the Z-basis measurement and Q_X is the error rate for the measurements in the X basis with respect to $|\mathrm{GHZ}\rangle$. Explicitly, these QBERs are defined by

$$Q_X = \frac{1 - \langle X^{\otimes N}\rangle}{2}, \tag{14}$$

$$Q_{AB_i} = \mathrm{Prob}\big(Z_A \neq Z_{B_i}\big) = \frac{1 - \big\langle Z_A Z_{B_i}\big\rangle}{2}. \tag{15}$$

The QBERs in general depend on the number of storage rounds of all parties (for the simulations, see Appendix A for details).

The asymptotic secret fraction r_∞, which is a central figure of merit, is also calculated. It defines the fraction of secret bits that can be extracted from the measured qubits. For the generalized N-BB84 protocol [3,4] the asymptotic secret fraction is

$$r_\infty\big(Q_X^{\mathrm{tot}}, Q_{AB_i}^{\mathrm{tot}}\big) = \max\left[0, 1 - h\big(Q_X^{\mathrm{tot}}\big) - \max_{1\leqslant i\leqslant N-1} h\big(Q_{AB_i}^{\mathrm{tot}}\big)\right], \tag{16}$$

with the binary Shannon entropy $h(p) = -p\log_2(p) - (1-p)\log_2(1-p)$. The total QBER $Q^{\mathrm{tot}} \in \{Q_X^{\mathrm{tot}}, Q_{AB_i}^{\mathrm{tot}}\}$ is given by the total number of measurements per round multiplied by the average QBER in each round divided by the total number of measurements summed over the whole number of rounds up to s_c:

$$Q^{\mathrm{tot}}(s_c) = \frac{\sum_{s=1}^{s_c}\langle l\rangle(s)\sum_{\delta_a}^{s}\cdots\sum_{\delta_{b_{N-1}}}^{s} Q\big(\delta_a,\ldots,\delta_{b_{N-1}}\big)\mathrm{Prob}(\delta_a)\cdots\mathrm{Prob}\big(\delta_{b_{N-1}}\big)(s)}{\sum_{s=1}^{s_c}\langle l\rangle(s)}. \tag{17}$$

Here $\mathrm{Prob}(\delta_i)$, with $i \in \{a, b_1, \ldots, b_{N-1}\}$, is the fraction of qubits of party i that are used in a GHZ measurement that has experienced decoherence for a certain number of rounds δ. The term $Q(\delta_a, \ldots, \delta_{b_{N-1}})$ is the theoretical QBER for the given storage times, which depend on all the fidelities of the stored qubits $a, b_1, \ldots, b_{N-1}$. The probability of white noise $p(\delta_i) = e^{-\delta_i/\tau}$ depends on all the storage rounds of all parties $\delta_a, \delta_{b_1}, \ldots, \delta_{b_{N-1}}$.

Finally, the secret key rate $K(s_c)$ can be calculated as the product of the asymptotic secret fraction from Eq. (16) and the router rate from Eq. (6):

$$K(s_c) = r_\infty\big(Q_X^{\mathrm{tot}}(s_c), Q_{AB_i}^{\mathrm{tot}}(s_c)\big)R(s_c). \tag{18}$$

In Appendix C an analytic calculation of the shared state after the GHZ measurement and the resulting quantum bit error rates is presented for the tripartite network. Using these results, we analyze different matching strategies with the aim of finding the one that optimizes the secret key rate. This comparison is presented in an exemplary way in the tripartite router setup in the following sections.

A. Strategies for storage and measurement

In [11], various strategies for how to choose memories for the BSMs were analyzed, with the aim of maximizing the secret key rate. Here we generalize the strategies to N parties. To integrate strategies into the matching process defined in Sec. III B, weights are associated with the edges, taking into account the number of storage rounds δ of the qubits that are included in a GHZ measurement. Two different ways are chosen to calculate the weights. (i) The weight W_1 of a hyperedge is defined by the absolute values of the difference in the number of storage rounds per two qubits, summed over all relevant bipartite edges within the hyperedge (see construction of the graph as in, e.g., Fig. 5). For the case $N = 3$ this reads

$$W_1 = |\delta_{b_1} - \delta_a| + \big|\delta_a - \delta_{b_2}\big|. \tag{19}$$

(ii) The weight W_2 is defined by the sum of the number of storage rounds of each qubit contained in the hyperedge. Explicitly, for the case $N = 3$,

$$W_2 = \delta_{b_1} + \delta_a + \delta_{b_2}. \tag{20}$$

Based on this, we compare the following strategies for the choice of the maximum matching among several possible ones.

Strategy S0. The first maximum matching found is chosen independently of the number of storage rounds.

Strategy S1. Taking the number of storage rounds of the qubits in the memories into account we (a) minimize over the sum of weights W_1 [defined in Eq. (19)] of all hyperedges and (b) maximize over the sum of weights W_1 [defined in Eq. (19)] of all hyperedges.

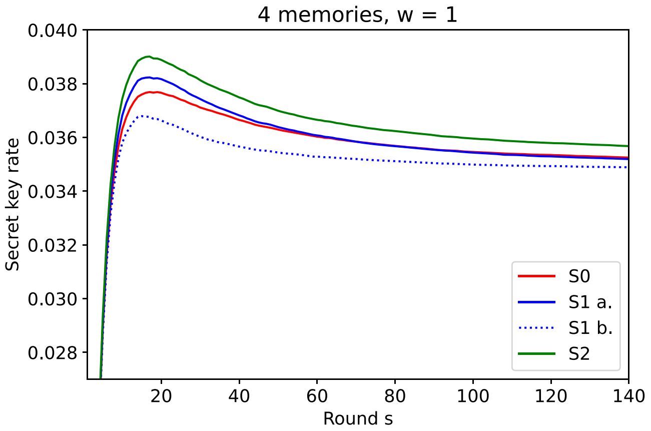


FIG. 9. Comparison of the secret key rate for the different strategies as described in Sec. IV A. In the tripartite network, each party has four memories and the maximal connection length is set to $w = 1$. The other parameters are $\eta = 0.1$, $\tau = 100$, and 50 000 samples.

Strategy S2. Taking the number of storage rounds of the qubits in the memories into account, we minimize the sum of weights W_2 [defined in Eq. (20)] of all hyperedges.

Strategy S1(a) produces states with the highest correlations by preferably choosing qubits with a minimal difference in the number of storage rounds, while qubits with large differences in the number of storage rounds are left over. As a consequence, older qubits are chosen less often than newer qubits. In contrast, older qubits are chosen earlier by combining them with the newest qubits in strategy S1(b). In strategy S2 we focus on a maximal fidelity by connecting qubits with the lowest number of storage rounds. Therefore, new qubits are measured as soon as possible. For example, with a memory allocation of $\delta_a = 0$ or 1, $\delta_{b_1} = 3$, and $\delta_{b_2} = 3$, strategy S1(a) would select the qubits with $\delta_a = 1$, $\delta_{b_1} = 3$, and $\delta_{b_2} = 3$, while strategy S2 would select the qubits with $\delta_a = 0$, $\delta_{b_1} = 3$, and $\delta_{b_2} = 3$ measuring newer qubits first.

Note that a maximization of the number of storage rounds (leading to a maximization of strategy S2) is not considered, as this increases the QBER and therefore leads to a decreasing secret key rate.

Figure 9 shows the secret key rate computed in a tripartite network with four memories per party. The maximal connection length is set to finite-range multiplexing ($w = 1$). The calculations are based on simulations with 50 000 samples to get the average number of GHZ measurements per round ($\langle l \rangle$) and the probability for the number of storage rounds [Prob(δ)]. Details of the simulation are given in Appendix A. The transmittivity is set to $\eta = 0.1$ and $\tau = 100$ is the decoherence parameter defined in Eq. (3). It turns out that considering the highest fidelities (i.e., the sum over the number of storage rounds has to be minimized; see strategy S2) leads to the highest secret key rate. Minimizing the difference in the number of rounds [see strategy S1(a)] leads to better results than choosing the first matching found (strategy S0) when performing only a few number of rounds. In the long-term, that changes and strategy S0 performs slightly better than strategy S1(a). Nevertheless, strategy S2 is the only strategy of practical use independent of the number of protocol rounds being performed.

Note that a maximal secret key rate is reached after about 16 rounds, as seen in Fig. 9. Exceeding this number of rounds, the secret key rate decreases due to the decoherence that the qubits experience in the memories. To reduce this effect, additional cutoffs will be introduced in a next step. Note that strategy S2 already leads to the idea of cutoffs, since older qubits are chosen with lower preference compared to new qubits.

B. Strategy: Emptying the memories after cutoffs

We now modify the protocol presented in Sec. III A by introducing cutoffs s_{cutoff} at the end of each round: Older qubits are removed from the memories such that they can be refilled in upcoming rounds. The previously defined strategy S2 of matching the newest qubits first is kept here and cutoffs are additionally considered.

The secret key rate plotted in Fig. 9 shows the competing behavior of the router rate and the asymptotic secret fraction. Due to the exponential decay of the fidelities that the qubits experience while being stored, a smaller cutoff leads to higher fidelity, which increases the asymptotic secret fraction. In contrast, the router rate decreases with smaller cutoffs, since a reduction in the cutoff comes along with a decreasing probability for memories to be filled. This can be seen in Figs. 10(a) and 10(b), where the router rate and the asymptotic secret fraction are plotted for different cutoffs. An optimal cutoff can be found where the two values multiply in a way that the secret key rate does not decrease with the number of rounds.

For our setup, such an optimal cutoff is achieved at $s_{\text{cutoff}} = 10$ rounds (seen by the simulations which were also done for $s_{\text{cutoff}} = 9$ and 11), which leads to the maximum secret key rate, as seen in Fig. 10(c). This cutoff can either be deduced from Fig. 9, where the secret key rate reaches roughly its maximum at about 16 rounds, or it can be argued mathematically via Eq. (16) for the secret fraction. Assuming that $Q_X = Q_{AB_i}$, none of the QBERs should be larger than 0.11 since otherwise $r_\infty = 0$. To achieve $Q_X \geqslant 0.11$, the mean fidelity that the qubits need to have when included in a GHZ measurement can be calculated. Here we assume that $F_A = 1$ and $F_{B_1} = F_{B_2}$ since one qubit is always new in the case of maximizing l. With the given parameters, $Q_X \geqslant 0.11$ is reached when the qubits are maximally stored for $s_{\text{cutoff}} = 12$ rounds. This is only an approximation since several assumptions are made here.

V. CONCLUSION

Quantum routers form a main ingredient when building quantum networks as they increase the communication distance between end users. By including quantum memories and multiplexing, the router rate (i.e., entanglement generation) and the secret key rate can be improved significantly. In this work, we presented a generalization of the bipartite repeater to the quantum router in an N-partite star graph with the quantum router being the central node. In contrast to previous work, we integrated quantum memories into the quantum router and considered multiplexing. We described the modified underlying graph-theoretic matching problem and implemented different algorithms for the maximum N-dimensional matching (with weights) to reduce the runtime as much as

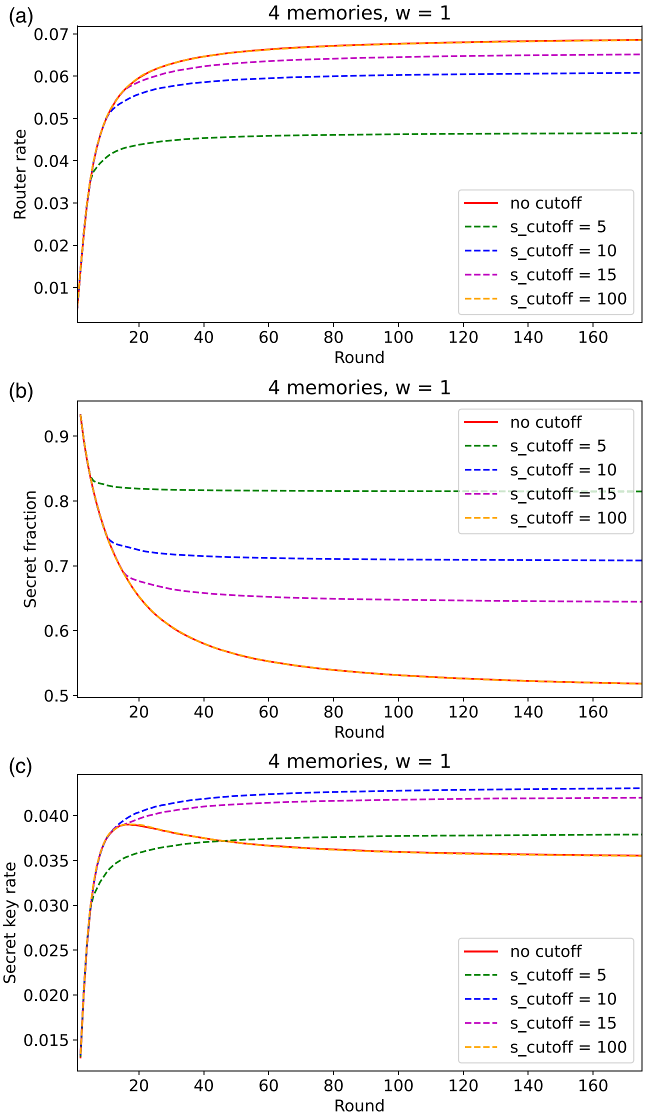


FIG. 10. Effect of the number of cutoff rounds on rates in a tripartite network with $m = 4$ memories per party and finite-range multiplexing $w = 1$. (a) Router rate for different cutoffs. (b) Secret fraction and (c) secret key rate for the same setup. The other parameters are $\eta = 0.1$, $\tau = 100$, and 50 000 samples.

possible. For $N = 3$, we designed an efficient algorithm in terms of using network flow.

Furthermore, we analyzed the router rate in star networks up to a network size of five parties with four memories each. The resulting plots show that even for finite-range multiplexing, large improvements in the router rate are achieved. In a second step, we considered conference key agreement and studied noise effects on the quantum memories. Due to decoherence, the asymptotic secret fraction, and therefore the secret key rate, decreases when qubits are stored longer. To minimize this effect, we investigated different storage and measurement strategies for the GHZ measurement. It turned out that it is best to connect the qubits with the shortest number of storage rounds (highest fidelities), i.e., to minimize the sum of the number of storage rounds in each matching. This strategy outperforms all other investigated strategies.

We further modified the protocol by removing older qubits from the memory after a certain number of rounds since they cannot contribute to the secret key rate. Combining the optimal matching strategy with the optimal number of cutoff rounds leads to the overall highest secret key rate.

In future work the analysis of larger networks with more than one central router will be of interest. Note that our analytical expressions hold for arbitrary N and m. The limiting factor in considering larger graphs is the underlying matching problem and the simulation of the protocol, which scale unfavorably with the network size. It should be investigated how the network structure influences the router rate and the secret key rate depending on multiplexing with different matching strategies. It might also be conceivable to extend the protocols and, for example, to take distillation into account. Regarding implementations, the influence of finite key effects might be integrated as well, following [21].

ACKNOWLEDGMENTS

We thank Dorothea Baumeister, Gláucia Murta, Jörg Rothe, and Nikolai Wyderka for valuable discussions. This work was funded by the Federal Ministry of Education and Research BMBF (Project QR.X with Project No. 16K1SQ023).

APPENDIX A: SIMULATION OF THE MULTIPLEXING PROTOCOL

To calculate the necessary parameters for the router rate and the secret key rate, we simulate the multipartite multiplexing protocol. To do so, the memories of each party are stored as an array that can take the value of either 0 (for empty memories) or an integer $i \in \mathbb{N}$, depending on the number of rounds the qubit has already been stored. All simulations are done in PYTHON. The steps of the simulation are as follows.

(i) Starting from the empty memory configuration, a decision is made for each memory in turn as to whether a qubit arrives and is stored or not. This happens randomly with a probability of η (given by the probability for a qubit to arrive at the router). If a qubit is stored, the corresponding array entry is set from 0 to 1. If the qubit is still in memory at the end of the round, the entry in the following round is increased by 1. In further rounds, a qubit is only sent if the memory is still empty.

(ii) Based on the memory configuration, the maximum matching is performed to decide which memories are involved in which GHZ measurement. Depending on the strategy chosen, as well as the number of parties and the connection length, the matching is implemented differently (see Appendix B).

The memory configuration determines how many GHZ measurements can be performed in each round [see Eq. (5)]. This results in the value for l. Furthermore, the probability of the average number of storage rounds Prob(δ_i) can be determined at this point considering each δ_i given by each qubit of the ith party.

(iii) The memories whose qubits were used in a GHZ measurement are set back to 0. Depending on whether a cutoff

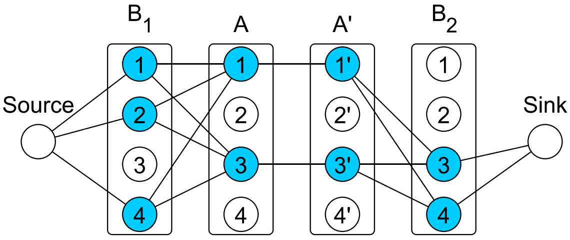


FIG. 11. Representation of the matching problem for the tripartite graph as a network flow. A source and a sink are added to the graph. All edges have a capacity of 1. To ensure that each party only appears once in a chosen flow (matching), it has to be ensured that each party only has one incoming or outgoing edge. Therefore, Alice's memory layer is copied and connected to the original layer accordingly. This ensures a single outgoing edge for A and a single incoming edge for A' resulting in the guaranty that A cannot be contained in two different flows (matchings).

is defined, all qubits whose number of storage rounds has exceeded the cutoff are removed from the memories as well.

(iv) Using this new memory configuration, the next round $s+1$ is started and the memory storage rounds are increased by 1 (if not empty).

The protocol is performed a fixed number of rounds. Additionally, the protocol is repeated several times, to get the average values for l and $\text{Prob}(\delta_i)(s)$.

APPENDIX B: MATCHING ALGORITHM

The matching is realized in different ways depending on the strategy, the number of parties, and the connection length.

For the tripartite network, the matching problem can be reduced to the well-known network flow problem [22], as the graph structure given by the circuit in Fig. 2 defines a flow from party B_1 over party A to party B_2. Assigning capacities to the edges and adding a source and a sink to the graph, the maximum flow from the source to the sink can be calculated. This corresponds to a maximum three-dimensional matching. As each node is allowed to appear in only one matching, the capacity has to be set to 1. Additionally, the array of the party in the middle (here party A) has to be doubled (layer A'). Edges are only allowed to be drawn between a node and its own copy (see Fig. 11). This has to be done to ensure that memories are not chosen more than once, as the input flow has to equal the output flow for each node. For party B_1 and party B_2 this is guaranteed, since these nodes are only connected to the source or sink on one side, respectively. In PYTHON, network flow can be realized using the `maximum_flow` function from `scipy.sparse.csgraph`.

For networks with more than three parties, the matching is realized as follows. At first, all memories that are filled in one round are included in the set of nodes V where the filled memories from one party form one subset V_i of vertices. In the next step, all valid edges and the resulting hyperedges between the memories from the disjoint subsets are identified depending on the connection length w. Then all possible combinations of hyperedges are considered sequentially. The first subset of hyperedges with no common vertices and maximum cardinality according to Eq. (5) that is found is chosen to be the matching. For parallel connections, i.e., $w=0$, there is only one matching that can be found, as no node has more than one incoming or outgoing edge, respectively. In the case of full-range multiplexing, i.e., $w=m-1$, a matching can be found by connecting the first filled memory from each party. Afterward, these nodes have to be erased from the graph so that they cannot be chosen a second time. This is repeated until one party has no more filled memories.

In the case of weighted matching, it is always necessary to find all allowed hyperedges (e.g., by combinations from the PYTHON package ITERTOOLS). From all combinations, the combination with maximum cardinality and maximum or minimum weight is chosen. This algorithm is also used for the tripartite network when weights are considered.

APPENDIX C: EXPLICIT CALCULATIONS FOR THE TRIPARTITE NETWORK

In the following, the calculations of the shared quantum state and the resulting quantum bit error rates are made explicitly for the case of the tripartite network. Starting from three Bell states $|\phi^+\rangle = \frac{1}{\sqrt{2}}(|00\rangle + |11\rangle)$ held by A, B_1, and B_2, the input state for the GHZ measurement can be calculated. We assume that the qubits stored in the memory undergo depolarization such that $|\phi^+\rangle\langle\phi^+|_i \rightarrow \rho_i^{\text{dep}} = F_i|\phi^+\rangle\langle\phi^+| + \frac{1-F_i}{3}(|\phi^-\rangle\langle\phi^-| + |\psi^+\rangle\langle\psi^+| + |\psi^-\rangle\langle\psi^-|)$ for $i \in \{A, B_1, B_2\}$. The input state is then given by the product state

$$\begin{aligned}\rho_{A_1B_1B_2}^{\text{dep}} &= \rho_{A_1}^{\text{dep}} \otimes \rho_{B_1}^{\text{dep}} \otimes \rho_{B_2}^{\text{dep}}\\ &= \rho_{a^{(1)}a^{(2)}}^{\text{dep}} \otimes \rho_{b_1^{(1)}b_1^{(2)}}^{\text{dep}} \otimes \rho_{b_2^{(1)}b_2^{(2)}}^{\text{dep}}.\end{aligned} \tag{C1}$$

The GHZ measurement is performed on the second qubit of the three parties ($a^{(2)}$, $b_1^{(2)}$, and $b_2^{(2)}$) according to the quantum circuit given in Fig. 2. The measurements performed at the end of the circuit are done in the Z basis. In the case of measuring three times a zero, the remaining first qubits ($a^{(1)}$, $b_1^{(1)}$, and $b_2^{(1)}$), held by each party, are projected onto the $|\text{GHZ}_0^+\rangle$ state (with a certain fidelity $\tilde{F} \leqslant 1$ because of the depolarization). Depending on the measurement outcome, a projection onto another GHZ state is also possible. In this case, the desired $|\text{GHZ}_0^+\rangle$ state can be achieved by the parties changing their local qubit according to $Z^{m_a^{(2)}} \otimes X^{m_{b_1}^{(2)}} \otimes X^{m_{b_2}^{(2)}}$ with the announced measurement outcomes $m_i^{(2)}$ of the second qubit, where $i \in \{a, b_1, b_2\}$. The corrected final state resulting from a GHZ measurement of the depolarized initial states is then given by the GHZ diagonal state

$$\begin{aligned}\tilde{\rho}_{A_1B_1B_2}^{\text{dep}} =& \lambda_0^+|\text{GHZ}_0^+\rangle\langle\text{GHZ}_0^+| + \lambda_0^-|\text{GHZ}_0^-\rangle\langle\text{GHZ}_0^-|\\ &+ \sum_{i=1}^{3}\lambda_i(|\text{GHZ}_i^+\rangle\langle\text{GHZ}_i^+| + |\text{GHZ}_i^-\rangle\langle\text{GHZ}_i^-|),\end{aligned} \tag{C2}$$

with the GHZ states $|\text{GHZ}_i^{\pm}\rangle = \frac{1}{\sqrt{2}}(|i\rangle \pm |N^2-1-i\rangle)$ and $i = 0, 1, \ldots, 2^{N-1}$ given in binary notation.

The GHZ diagonal elements $\lambda_i^{\pm}$ are given by

$$\begin{aligned}\lambda_0^+ = \tfrac{1}{27}\big(&4 - F_A - F_{B_1} - F_{B_2} - 2F_AF_{B_1}\\ &- 2F_{B_1}F_{B_2} - 2F_AF_{B_2} + 32F_AF_{B_1}F_{B_2}\big),\end{aligned} \tag{C3}$$

$$\begin{aligned}\lambda_0^- = \tfrac{1}{27}\big(5 - 5F_A - 5F_{B_1} - 5F_{B_2} + 14F_AF_{B_1}\\ + 14F_{B_1}F_{B_2} + 14F_AF_{B_2} - 32F_AF_{B_1}F_{B_2}\big),\end{aligned} \tag{C4}$$

$$\lambda_1 = \tfrac{1}{9}\big(F_{B_2} + 2F_AF_{B_1} - 2F_AF_{B_2} - 2F_{B_1}F_{B_2} + 1\big), \tag{C5}$$

$$\lambda_2 = \tfrac{1}{9}\big(F_{B_1} - 2F_AF_{B_1} - 2F_{B_1}F_{B_2} + 2F_AF_{B_2} + 1\big), \tag{C6}$$

$$\lambda_3 = \tfrac{1}{9}\big(F_A - 2F_AF_{B_1} - 2F_AF_{B_2} + 2F_{B_1}F_{B_2} + 1\big), \tag{C7}$$

with initial fidelity $F_i = \frac{1}{4} + \frac{3}{4}e^{-\delta_i/\tau}$ for each party $i \in \{A, B_1, \dots, B_{N-1}\}$. Note that one of these fidelities has to be 1 in each round since we maximize the number of GHZ measurements l. The fidelity of the state after correction according to the measurement outcome is

$$\tilde{F}_{\tilde{\rho}^{\rm dep}} = \langle \mathrm{GHZ}_0^+ | \tilde{\rho}^{\rm dep}_{a^{(1)}b_1^{(1)}b_2^{(1)}} | \mathrm{GHZ}_0^+ \rangle = \lambda_0^+. \tag{C8}$$

For the QBER in the X basis, it holds that

$$Q_X = \frac{1 - \langle X^{\otimes N}\rangle}{2} = \frac{1 - (\lambda_0^+ - \lambda_0^-)}{2}. \tag{C9}$$

The bipartite QBERs are given by

$$\begin{aligned}Q_{AB_1} &= 2(\lambda_2 + \lambda_3),\\ Q_{AB_2} &= 2(\lambda_1 + \lambda_3).\end{aligned} \tag{C10}$$

With the QBERs from Eqs. (C9) and (C10), the total error ($Q_X^{\rm tot}$ and $Q_{AB_i}^{\rm tot}$) over the protocol (all rounds up to a current round s_c) between all outcomes of A and B_i can then be calculated following Eq. (17):

$$Q^{\rm tot}(s_c) = \frac{\sum_{s=1}^{s_c} \langle l\rangle(s) \sum_{\delta_{a^{(2)}}}^{s} \sum_{\delta_{b_1^{(2)}}}^{s} \sum_{\delta_{b_2^{(2)}}}^{s} Q\big(\delta_{a^{(2)}}, \delta_{b_1^{(2)}}, \delta_{b_2^{(2)}}\big)\mathrm{Prob}(\delta_{a^{(2)}})\mathrm{Prob}(\delta_{b_1^{(2)}})\mathrm{Prob}\big(\delta_{b_2^{(2)}}\big)(s)}{\sum_{s=1}^{s_c} \langle l\rangle(s)}. \tag{C11}$$

With this, we can finally compute the asymptotic secret fraction from Eq. (16) and further the secret key rate given in Eq. (18).

[1] C. H. Bennett and G. Brassard, *Proceedings of the IEEE International Conference on Computers, Systems and Signal Processing, Bangalore* (IEEE, Piscataway, 1984), pp. 175–179.

[2] A. K. Ekert, Quantum cryptography based on Bell's theorem, Phys. Rev. Lett. **67**, 661 (1991).

[3] G. Murta, F. Grasselli, H. Kampermann, and D. Bruß, Quantum conference key agreement: A review, Adv. Quantum Technol. **3**, 2000025 (2020).

[4] F. Grasselli, H. Kampermann, and D. Bruß, Finite-key effects in multipartite quantum key distribution protocols, New J. Phys. **20**, 113014 (2018).

[5] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, The security of practical quantum key distribution, Rev. Mod. Phys. **81**, 1301 (2009).

[6] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, Quantum repeaters: The role of imperfect local operations in quantum communication, Phys. Rev. Lett. **81**, 5932 (1998).

[7] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, Quantum repeaters based on entanglement purification, Phys. Rev. A **59**, 169 (1999).

[8] Z.-D. Li, R. Zhang, X.-F. Yin, L.-Z. Liu, Y. Hu, Y.-Q. Fang, Y.-Y. Fei, X. Jiang, J. Zhang, L. Li, *et al.*, Experimental quantum repeater without quantum memory, Nat. Photon. **13**, 644 (2019).

[9] S. Abruzzo, H. Kampermann, and D. Bruß, Measurement-device-independent quantum key distribution with quantum memories, Phys. Rev. A **89**, 012301 (2014).

[10] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, Multiplexed memory-insensitive quantum repeaters, Phys. Rev. Lett. **98**, 060502 (2007).

[11] S. Abruzzo, H. Kampermann, and D. Bruß, Finite-range multiplexing enhances quantum key distribution via quantum repeaters, Phys. Rev. A **89**, 012303 (2014).

[12] H.-K. Lo, M. Curty, and B. Qi, Measurement-device-independent quantum key distribution, Phys. Rev. Lett. **108**, 130503 (2012).

[13] M. Epping, H. Kampermann, and D. Bruß, Large-scale quantum networks based on graphs, New J. Phys. **18**, 053036 (2016).

[14] T. Coopmans, R. Knegjens, A. Dahlberg, D. Maier, L. Nijsten, J. de Oliveira Filho, M. Papendrecht, J. Rabbie, F. Rozpędek, M. Skrzypczyk, L. Wubben, W. de Jong, D. Podareanu, A. Torres-Knoop, D. Elkouss, and S. Wehner, NetSquid, a network simulator for quantum information using discrete events, Commun. Phys. **4**, 164 (2021).

[15] G. Vardoyan, S. Guha, P. Nain, and D. Towsley, On the stochastic analysis of a quantum entanglement distribution switch, IEEE Trans. Quantum Eng. **2**, 1 (2021).

[16] C.-L. Li, Y. Fu, W.-B. Liu, Y.-M. Xie, B.-H. Li, M.-G. Zhou, H.-L. Yin, and Z.-B. Chen, Breaking universal limitations on quantum conference key agreement without quantum memory, Commun. Phys. **6**, 122 (2023).

[17] R. M. Karp, in *Complexity of Computer Computations: Proceedings of a Symposium on the Complexity of Computer Computations, Yorktown Heights*, edited by R. E. Miller, J. W. Thatcher, and J. D. Bohlinger (Springer, Boston, 1972), pp. 85–103.

[18] J. Hartmanis, Computers and intractability: A guide to the theory of NP-completeness (Michael R. Garey and David S. Johnson), SIAM Rev. **24**, 90 (1982).

[19] J. A. Kunzelmann, A. Trushechkin, N. Wyderka, H. Kampermann, and D. Bruß (unpublished).

[20] D. A. Levin and Y. Peres, *Markov Chains and Mixing Times* (American Mathematical Society, Providence, 2017), Vol. 107.

[21] T.-T. Song, Q.-Y. Wen, F.-Z. Guo, and X.-Q. Tan, Finite-key analysis for measurement-device-independent quantum key distribution, Phys. Rev. A **86**, 022332 (2012).

[22] A. V. Goldberg, É. Tardos, and R. Tarjan, Network flow algorithms, Cornell University Report No. CS-TR-216-89, 1989 (unpublished).

Paper C

Multiplexed multipartite quantum repeater rates in the stationary regime

Julia A. Kunzelmann,[1, *] Anton Trushechkin,[1, 2, *] Nikolai Wyderka,[1] Hermann Kampermann,[1] and Dagmar Bruß[1]

[1] *Institute for Theoretical Physics III, Heinrich Heine University Düsseldorf, D-40225 Düsseldorf, Germany*
[2] *Steklov Mathematical Institute of Russian Academy of Sciences, Gubkina Str. 8, 119991 Moscow, Russia*

Multipartite quantum repeaters play an important role in quantum communication networks enabling the transmission of quantum information over larger distances. To increase the rates for multipartite entanglement distribution, multiplexing of quantum memories is included. Understanding the limitations of achievable rates in the stationary regime for different network sizes is a fundamental step to comprehend scalability of quantum networks. This work investigates the behavior of the multipartite quantum repeater rate (i.e., the number of GHZ states generated per round and per memory) in the stationary regime in multipartite star graphs with a single central multipartite quantum repeater including multiplexing using Markov chains. We derive a closed-form expression for the stationary rate depending on the network size. We support our results with numerical simulations. Further, we show that the rate saturates for large number of memories. On an abstract level, the mathematical description is equivalent to quantum repeater chains between two parties. Therefore, our results also apply to those setups.

I. INTRODUCTION

Quantum communication over large distances in quantum networks relies on the availability of entangled quantum pairs between end nodes. By increasing the distance between two nodes, the success probability for distributing entangled quantum states decreases [1]. To overcome this problem, quantum repeaters have been suggested [2–4]. By performing Bell state measurements in a central quantum repeater, entanglement between two distant parties can be generated. Analogously, in the multipartite setup, a GHZ measurement can be performed to entangle all n parties that are connected with the multipartite quantum repeater [5]. To further increase the rate of entanglement distribution in quantum networks, multiplexing can be introduced to the quantum repeater [6], where each party has m parallel quantum channels with the repeater. Instead of sending one qubit to the repeater, m qubits can be sent in parallel. Consequently, up to m entangling operations can be performed in parallel at the central station, such that the entanglement distribution rate is increased. This has been analyzed theoretically for the bipartite [7] as well as the multipartite setup [8]. Since multiplexing is related to an increased cost of resources, understanding the scaling of such multiplexed multipartite quantum repeaters with respect to the number of parties and memories per party is of great interest.

A repeater chain of n segments with $n-1$ repeater stations between two parties was considered in Ref. [9]. By using Markov chains, the authors calculated the average waiting times and, based on this, the transmission rates. In Ref. [10], also the average waiting time was analyzed for evaluating entanglement distribution in quantum networks. The entanglement generation rate was analyzed in Ref. [11]. In Ref. [12], the secret key rate for repeater chains with more segments was computed. The authors additionally considered more realistic setups, including memory cutoffs and repeater chains running in parallel. Bipartite quantum repeaters with multiplexing were studied in Ref. [13–15].

In this paper, we investigate multipartite quantum repeaters that connect more than two parties based on the setup shown in Fig. 1. Here, one multipartite quantum repeater is placed between n parties, that are connected to a central station. Each party has m memories available at the multipartite quantum repeater. These memories allow the parties to send m qubits in parallel. The purpose of the multipartite quantum repeater is to generate Greenberger-Horne-Zeilinger (GHZ) multipartite entangled states. As a figure of merit, we consider the average number of GHZ states generated per round and per memory, called “multipartite repeater rate”. In the following, we simply denote it as repeater rate.

In contrast to previous works, we are not interested in the average waiting time for generating the desired state, starting from the initial state of empty memories, but in the repeater rate of the stationary (i.e., long-term) regime. The difference with the scenario of Ref. [12] is that, after a successful GHZ measurement, we do not empty all multiplexed memories, but keep them for future rounds. In other words, the next round does not start again from the initial state, but starts from a state where some memories can be already filled. Then we obtain a random process (a Markov chain) and are interested in its stationary regime, namely in stationary repeater rate.

The problem of finding the stationary state of a Markov chain is reduced to the solution of a system of linear equations. It can be solved numerically, but it is preferable to have an analytic solution for understanding the influences of various parameters on the performance of such systems. Unfortunately, this system is intractable analytically, even for a moderate number of parties and memories. So, suitable approximations are required, which we use to derive explicit formulas that

[*] These two authors contributed equally

give a very good approximation for the repeater rate for an arbitrary number of parties and memories.

It is worth noting that, on the considered idealized level of description, our scenario is mathematically equivalent to a transmission line between two participants, Alice and Bob, with $n-1$ intermediate quantum repeaters that split the transmission line into n segments and generate Bell pairs on each segment. Thus, though we will focus on the case of a multipartite quantum repeater between n participants, our results can be applied also to a chain of $n-1$ quantum repeaters between two participants.

Our paper is structured as follows. In Sec. II, we introduce the n-partite quantum repeater with multiplexing and its description via Markov chains. In Sec. III, we analyze the setup without multiplexing and show the equivalence to the setup of a bipartite repeater chain. We move on to the generalized multipartite quantum repeater including multiplexing in Sec. IV. We give bounds on the repeater rate in the multipartite repeater setup, including multiplexing. We further discuss the effect of memories on the repeater rate and the dependency on the number of parties as well as the number of memories in larger networks. In Sec. V, we present our conclusion and outlook.

II. THE SETUP

A. Physical description

We consider a star network with one central multipartite quantum repeater and n parties around it [8] (see Fig. 1). Each party has $m \geq 1$ sources producing Bell pairs. We will refer to the case $m > 1$ as multiplexing. Additionally, each party has m quantum memories in the multipartite quantum repeater to store arriving qubits [6]. For simplicity, we assume that the memories are perfect and have infinite storage time. Each source sends one qubit of the Bell pair via a quantum channel to the multipartite quantum repeater per unit of time (round). The qubit is stored and heralded if it arrives successfully (with probability p). If each party has at least l filled memories, then the parties can perform l GHZ measurements. Fig. 1 shows a configuration allowing two GHZ measurements.

We consider the case of no restriction on the coupling of memory cells inside the multipartite repeater, i.e., if each party has at least one filled memory, a GHZ measurement can be performed independently of the positions of the filled memories in the memory stacks of each party. The case of restrictions was analyzed in Ref. [8].

Recall that the n-partite GHZ measurement is the measurement corresponding to the orthonormal basis

$$\{X_1^{b_1} \dots X_{n-1}^{b_{n-1}} Z_n^{b_n} |\mathrm{GHZ}\rangle\}, \qquad (1)$$

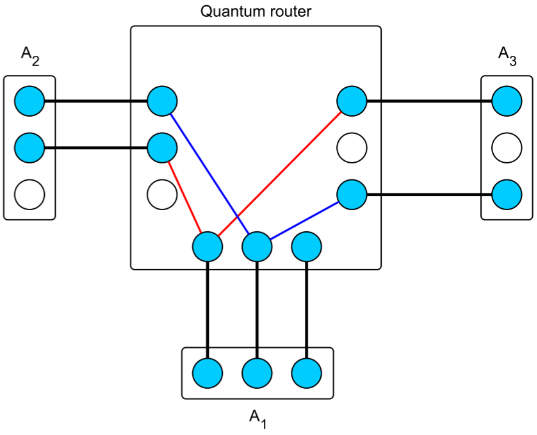


Figure 1. Setup of a tripartite multipartite quantum repeater with $n = 3$ parties and $m = 3$ memories per party adapted from [8]. The blue and white vertices (balls) correspond to empty and filled quantum memory cells, respectively. The edges between the parties and the multipartite quantum repeater (depicted in black) correspond to established Bell pairs. Edges inside the multipartite quantum repeater (depicted in red and blue) correspond to possible combinations allowing two GHZ measurements.

$b_1, \dots, b_n \in \{0, 1\}$, where

$$|\mathrm{GHZ}\rangle = \frac{1}{\sqrt{2}}(|0\rangle^{\otimes n} + |1\rangle^{\otimes n}) \qquad (2)$$

and X_i and Z_i are the Pauli operators acting on the ith qubit. Each state from this basis (i.e., a postmeasurement state) is local-unitary equivalent to the GHZ state, and the local unitaries are determined by the announced measurement outcome. We assume that the GHZ measurements are perfect.

The GHZ states are then used in an application, e.g., in a conference key agreement protocol [16]. Thus, memories that were included in a measurement are emptied again, see Fig. 2. All other memories remain untouched for the next round. In the following, qubits from a Bell pair are only sent to memories that are not filled from a previous round.

The goal of the protocol is to perform as many GHZ measurements l as possible per round and consequently maximize the rate in the stationary regime. More precisely, the number of GHZ measurements is a random variable L (all random variables will be denoted as capital Latin letters, and their possible values will be denoted as the corresponding small letters) because the storage process is probabilistic. We want to maximize its average value $\langle L \rangle$ in the stationary regime.

B. States of the multipartite quantum repeater and transitions

Let us describe the set of configurations of the memory cells in the multipartite quantum repeater. The multi-

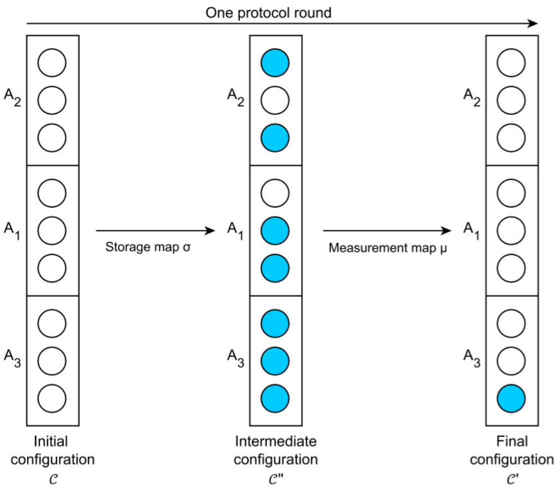


Figure 2. Schematic representation of a protocol round analogous to [7, 8]. Shown are the memory configurations for a multipartite quantum repeater with $n = 3$ parties, each with $m = 3$ memories. The transition of the memory configurations consists of the two processes of sending/storing (map σ) and measuring (map μ).

partite repeater has m qubit memory cells for each party, i.e., nm memory cells in total. Each party also has m memory cells, which are in one-to-one correspondence with the corresponding multipartite repeater's memory cells, see Fig. 1. Thus, each party's memory cell is assigned to its own multipartite repeater memory cell and one can speak about pairs of memory cells. Since the memories are filled by pairs, we will use the notions of pairs of memory cells or single memory cells (from the party or repeater side) interchangeably.

Thus, the configuration of memories in the whole system is uniquely defined by the memory configuration in the multipartite repeater. The configuration of the pairs of memories for the ith party can be defined as a binary vector $\vec{a}_i = (a_{i1}, a_{i2}, \dots, a_{im})$ with length m. Here, $a_{ij} = 0$ and $a_{ij} = 1$ correspond to an empty and filled ith pair of memories, respectively. The total memory configuration is thus given by $\mathbf{a} = (\vec{a}_1, \vec{a}_2, \dots, \vec{a}_n) \in \{0, 1\}^{nm}$.

Since the positions of the filled memories are unimportant for the possibility of the GHZ measurement, we are actually interested in the number of filled memories $|\vec{a}_i|$ (here, $|\cdot|$ denotes the Hamming weight of a binary vector, i.e., the number of ones) for each party rather than the vectors $\vec{a}_i$ themselves. This can be used in simulations to reduce the configuration space, and we will use this more economic description in Sec. IV. However, in this section, it will be conceptually more convenient to use the vectors $\vec{a}_i$.

Each round of the multipartite quantum repeater includes two steps, depicted in Fig. 2. The first step is the storage step. A Bell state source is assigned to each of the nm pairs of memories between the parties and the multipartite repeater. During the storage step, for each empty pair of memories, the corresponding Bell state source tries to establish a Bell pair and store it in the pair of memories. The success probability of this single event is p. Possible reasons for the failure of Bell-state generation are discussed, e.g., in Ref. [17]. Note that the success of a Bell state generation is considered independently for each link.

The second step is called "measurement" which corresponds to emptying some memories due to the GHZ measurement. Namely, if each party has at least l filled memories and one party has exactly l filled memories, then l GHZ measurements are performed and l memories are emptied for each party. The precise positions of the emptied memories are unimportant. For definiteness, let us assume that each party tries to empty memories with higher indices first.

To answer the question of how many GHZ measurements can be performed per round in the stationary regime, we will employ the theory of Markov chains.

C. Description via Markov chains

Let $\pi_{\mathbf{a}}$ denote the probability for the system to be in the configuration $\mathbf{a} \in \{0, 1\}^{nm}$ in some moment of time. The probability distribution $\{\pi_{\mathbf{a}}\}_{\mathbf{a} \in \{0,1\}^{nm}}$ can be represented as a vector π of length nm. Then the two steps described in the previous subsection correspond to $2^{nm} \otimes 2^{nm}$ transition matrices σ (for the storage step) and μ (for the measurement step) between the configurations. The (stochastic) transition matrix T of one "working cycle", or round, is then given by

$$T = \mu\sigma. \tag{3}$$

A matrix element $T_{\mathbf{a}'\mathbf{a}}$ gives the probability of changing the multipartite quantum repeater's state from any configuration $\mathbf{a}$ to any configuration $\mathbf{a}'$ (i.e., the column contains the actual configuration and the row the future configuration).

As an example, consider the case of $n = 2$ and $m = 1$, i.e., two participants, each with a single memory. As soon as both memories are filled with a quantum bit, both memories are emptied by performing a Bell state measurement. In this case, $\vec{a}_1$ and $\vec{a}_2$, are simply single bits and we can write $\mathbf{a} = (a_1, a_2)$.

Such a Markov chain has four different configurations: $\mathbf{a} = (0, 0)$ (both memories in the repeater are empty), $\mathbf{a} = (0, 1)$ and $\mathbf{a} = (1, 0)$ (one memory is filled and one is empty) and $\mathbf{a} = (1, 1)$ (both memories are filled). The storage map $\sigma^{(1)}$ for one party has the form ,

$$\sigma^{(1)} = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}. \tag{4}$$

Here, the first and second rows/columns correspond to $a_i = 0$ (empty memory) and $a_i = 1$ (filled memory),

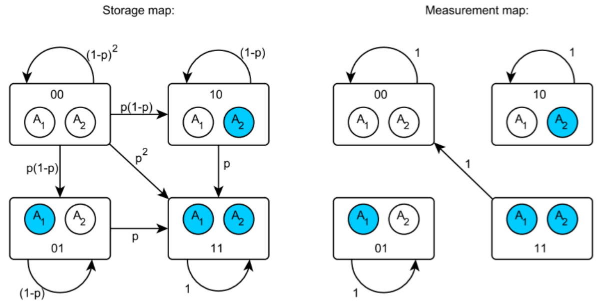


Figure 3. Graph representation of the storage map σ (left) and the measurement map μ (right) of a bipartite quantum repeater with one memory per party. The vertices correspond to configurations, and the edges correspond to possible transitions between them. Along the edges, the transition probabilities depending on the success probability p are given. The Bell measurement is considered to be perfect.

respectively. The first column indicates that an empty memory remains empty with probability $1-p$ and becomes filled with probability p. The second column means that the filled memory remains filled with probability 1 in this stage. The total storage map σ is obtained via the tensor product:

$$\sigma = \sigma^{(1)} \otimes \sigma^{(1)} = \begin{pmatrix} (1-p)^2 & 0 & 0 & 0 \\ p(1-p) & 1-p & 0 & 0 \\ p(1-p) & 0 & 1-p & 0 \\ p^2 & p & p & 1 \end{pmatrix}, \quad (5)$$

where the rows and columns (corresponding the bipartite configurations $\mathbf{a}$) are ordered in the usual way: $(0,0)$, $(0,1)$, $(1,0)$, $(1,1)$. The measurement map μ is as follows:

$$\mu = \begin{pmatrix} 1 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (6)$$

That is, the memory configuration changes in this step only if both memories are filled and a Bell state measurement can be performed. A graph representation of both maps is shown in Fig. 3. The transition matrix T is then given by

$$T = \mu\sigma = \begin{pmatrix} (1-p)^2 + p & p & p & 1 \\ p(1-p) & 1-p & 0 & 0 \\ p(1-p) & 0 & 1-p & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}. \quad (7)$$

Returning from our example, the stationary distribution π^* of a Markov chain is a distribution π (represented as a column vector) that does not change when the transition matrix T is applied:

$$\pi^* = T\pi^*. \quad (8)$$

The stationary distribution can also be calculated by taking the limit of T^s with infinitely many rounds s and applying it to an initial distribution $\pi^{(0)}$:

$$\pi^* = \lim_{s\to\infty} T^s \pi^{(0)}. \quad (9)$$

Note that π^* is the stationary probability distribution for memory configurations at the end of a round, i.e., after a measurement step. The number of GHZ state measurements in any round is specified by the configuration before this step, i.e., after the storage map. So, to get the average number of GHZ state measurements, we need to apply the storage map, i.e., we need to consider the vector $\sigma\pi^*$.

The number of GHZ measurements in the stationary regime as a random variable is denoted by L. The average value of the number of the GHZ measurement $\langle L\rangle$ is completely determined by $\sigma\pi^*$:

$$\langle L\rangle = \sum_{\mathbf{a}\in\{0,1\}^{nm}} (\sigma\pi^*)_{\mathbf{a}} \min_{i=1,\dots,n} |\vec{a}_i|, \quad (10)$$

where $(\sigma\pi^*)_{\mathbf{a}}$ denotes the entry of the vector $\sigma\pi^*$ and, recall, $|\cdot|$ denotes the Hamming weight of the vector. The minimal Hamming weight over all parties is the number of the GHZ measurements l for a given configuration. From that, the asymptotic repeater rate per memory can be calculated as

$$R_\infty = \frac{\langle L\rangle}{m} \quad (11)$$

with m being the number of memories per party.

The Markov chain that represents the transitions of a multipartite quantum repeater during one round is irreducible. A Markov chain is irreducible if any configuration can reach all other configurations (within a finite number of rounds). This is true in the setup of the multipartite quantum repeater, as any number of memories can be filled in every round, and the configuration with all memories filled can always be reached, allowing them to be emptied again. Therefore, it is always possible to reach any configuration $\mathbf{a}'$ from any configuration $\mathbf{a}$ within a finite number of rounds. The stationary distribution of an irreducible Markov chain is unique, i.e., it is independent of the initial distribution $\pi^{(0)}$. A natural choice for $\pi^{(0)}$ is the case of all memories being empty.

III. MULTIPARTITE QUANTUM REPEATER WITHOUT MULTIPLEXING

First, we consider a multipartite quantum repeater with a single memory per party, i.e., without multiplexing. In this case, $\mathbf{a} = (a_1, \dots, a_n)$, where a_i are bits corresponding to the states of the single memories of each party. A GHZ measurement is possible only in the configuration $\mathbf{a} = (1, \dots, 1)$, i.e., the memory for each party is filled. The average number of GHZ measurements per round, or, equivalently, for this case, the probability of performing a single GHZ measurement in the stationary regime, is given by

$$\langle L_1 \rangle = \Pr(L_1 = 1) = (\sigma\pi^*)_{(1,\ldots,1)}, \tag{12}$$

where the subscript 1 in the notation L_1 denotes that we consider the single-memory (per party) case. It turns out that

$$\langle L_1 \rangle = \left[1 + \sum_{j=1}^{\infty} \left(1 - \left(1 - (1-p)^j \right)^n \right) \right]^{-1}. \tag{13}$$

The derivation of Eq. (13) is provided in Appendix A, but this result (in a different form) was already known in the context of chains of quantum repeaters between two participants [17]. To show this, let us simplify Eq. (13):

$$\begin{aligned}\langle L_1 \rangle &= \left[1 + \sum_{j=1}^{\infty} \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} (1-p)^{jk} \right]^{-1} \\ &= \left[1 + \sum_{k=1}^{n} (-1)^{k+1} \binom{n}{k} \frac{(1-p)^k}{1-(1-p)^k} \right]^{-1} \\ &= \left[\sum_{k=1}^{n} \frac{(-1)^{k+1}}{1-(1-p)^k} \binom{n}{k} \right]^{-1}. \end{aligned} \tag{14}$$

The last expression in the square brackets is well-known in probability theory: It is the expectation value of the maximum of n independent geometrically distributed random variables with the success probability p [18]. In our context, this represents the average waiting time for the GHZ measurement when starting from empty memories. Indeed, the geometric distribution is the probability distribution of the number of Bernoulli trials (random experiments with exactly two possible outcomes) to get one success. For a successful GHZ measurement, all n parties must have a filled memory, hence the waiting time is the maximum number of Bernoulli trials over all participants.

Then, $\langle L_1 \rangle$ is the inverse of the expectation value of the maximum of n independent geometrically distributed random variables. This is due to the absence of multiplexing: Once a GHZ measurement is performed, the system returns to the initial state with all memories empty.

The derivation in Appendix A can be considered as an alternative derivation of the expectation value of the maximum of n independent geometrically distributed random variables. The advantage of Eq. (14) is that the denominator contains a finite sum, but the terms have alternating signs. The advantage of Eq. (13) is that all terms have the same sign.

The expression in the square brackets in Eq. (14) for the waiting time was obtained in a model of two parties connected by a sequence of n segments with $n-1$ repeaters [17] (see also Refs. [9, 19]), which is, as we noticed above, mathematically equivalent to the considered model of a multipartite quantum repeater.

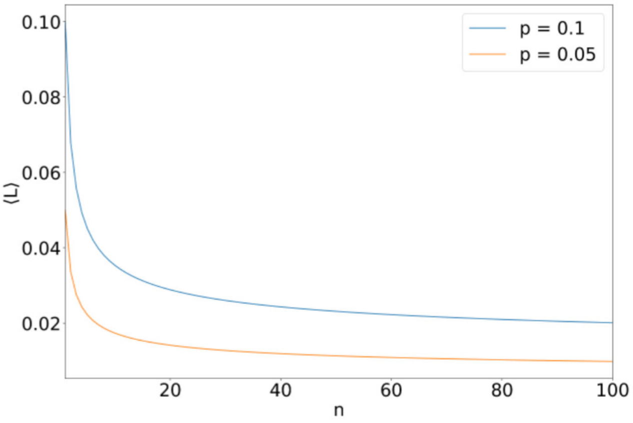


Figure 4. Multipartite quantum repeater without multiplexing: average number of GHZ measurements performed in the asymptotic limit following Eq. (13) for two different success probabilities $p = 0.1$ and $p = 0.05$, as a function of the number of parties n.

Known estimates on the expectation value of the maximum of n independent geometrically distributed random variables are [20]:

$$\frac{1}{\ln \frac{1}{1-p}} \sum_{k=1}^{n} \frac{1}{k} \leq \langle L_1 \rangle^{-1} \leq 1 + \frac{1}{\ln \frac{1}{1-p}} \sum_{k=1}^{n} \frac{1}{k}. \tag{15}$$

Recall that

$$\sum_{k=1}^{n} \frac{1}{k} = \ln n + \gamma + \frac{1}{2n} + O(n^{-2}), \tag{16}$$

where $\gamma = 0.57721\ldots$ is the Euler-Mascheroni constant. Substitution of Ineqs. (15) into Eq. (14) leads to the bounds for the stationary repeater rate in the case of no multiplexing. Thus, the rate $\langle L_1 \rangle$ decreases slowly as $(\ln n)^{-1}$ for a large number of communicating parties n.

Fig. 4 shows the rates for two different success probabilities. One can see that, although the decrease in the average repeater rate is slow for a large number of communicating parties n, it decreases rapidly for small n. This motivates the use of multiple memories per party, allowing multiplexing to be integrated into the key distribution protocol [7, 8], which is discussed in the following section.

IV. THE CASE OF MULTIPLEXING

A. Reduction of the configuration space

Now consider the case of memory multiplexing, i.e., $m \geq 2$. We already mentioned that the number of GHZ measurements depends only on the numbers of the filled memories for each party $|\vec{a}_i|$ [see Eq. (10)], but not on their positions. Hence, we can merge the configurations corresponding to the same tuples $(|\vec{a}_i|)_{i=1}^n$. We will denote $|\vec{a}_i| = k_i$ and $(k_1, \ldots, k_n) = \mathbf{k}$. Then the number of configurations is reduced from 2^{nm} to $(m+1)^n$: Each of the n parties is fully characterized by the number of filled memories, from 0 to m.

Then we can interpret π as a vector of length $(m+1)^n$, and σ, μ, and $T = \mu\sigma$ as $(m+1)^n \times (m+1)^n$ matrices.

The storage map is given by the tensor power $\sigma = (\sigma^{(1)})^{\otimes n}$, where the single-particle storage map is given by

$$\sigma^{(1)}_{k'k} = \begin{cases} 0, & k > k', \\ \binom{m-k}{k'-k}(1-p)^{(m-k')}p^{(k'-k)}, & \text{otherwise,} \end{cases} \tag{17}$$

with $k, k' \in \{0, 1, \ldots, m\}$. The matrix of the measurement map is defined as follows: For arbitrary $\mathbf{k} = (k_1, \ldots, k_n)$ and $\mathbf{k}' = (k'_1, \ldots, k'_n)$, where $k_i, k'_i \in \{0, \ldots, m\}$,

$$\mu_{\mathbf{k}'\mathbf{k}} = \begin{cases} 1, & \mathbf{k} - \mathbf{k}' = l \cdot \vec{1} \text{ and } \min \mathbf{k}' = 0, \\ 0, & \text{otherwise,} \end{cases} \tag{18}$$

where $\vec{1} = (1, \ldots, 1)$ is the vector containing n ones, $\min \mathbf{k}'$ is the minimum over $k'_1, \ldots, k'_n$, and $l \in \{0, \ldots, m\}$ is the number of GHZ measurements. Then Eq. (10) becomes

$$\langle L \rangle = \sum_{\mathbf{k}=\{0,\ldots,m\}^n} (\sigma\pi^*)_{\mathbf{k}}(\min \mathbf{k}). \tag{19}$$

We can apply one more reduction of the configuration space if we merge configurations that differ only by permutations of the parties. That is, $\mathbf{k} = (k_1, \ldots, k_n)$ and $\mathbf{k}' = (k'_1, \ldots, k'_n)$ are considered equivalent if $(k'_1, \ldots, k'_n)$ can be obtained from $(k_1, \ldots, k_n)$ by a permutation of elements. Then, we can define a "canonical" order of k_i as, e.g., decreasing: $k_1^\downarrow \geq k_2^\downarrow \geq \ldots \geq k_n^\downarrow$ and the state is described by an ordered n-tuple $(k_1^\downarrow, \ldots, k_n^\downarrow)$.

B. Bipartite setup ($n = 2$) for small success probabilities p

For the bipartite case (i.e., $n = 2$) with multiplexing [7], a general expression can be deduced under the assumption of a small Bell pair generation probability $p \ll 1$. Then, all formulas are approximated by the first order in p, e.g., $(1-p)^m \approx 1 - mp$. Physically, that means that at most one memory can be filled per round. Nevertheless, qubits are kept in the memories over rounds once they are stored. The advantage is that the transition matrix has fewer entries that are unequal to 0. Consequently, the graph representing the Markov chain reduces significantly (see Fig. 5).

We use the description of the configuration space from the end of Sec. IV A. Namely, a configuration is a pair $(k_1^\downarrow, k_2^\downarrow)$ of parties' occupation numbers in decreasing order. For example, the vertex $(1, 0)$ includes all configurations where one party has one filled memory. Since we empty memories (by performing the GHZ measurement) at the end of a round, configurations where both parties have filled memories, such as $(1, 1)$, cannot be reached after each iteration, as it ends with a measurement step. Therefore, these configurations do not appear in the graph shown in Fig. 5. Including all assumptions, the number of vertices, each representing one configuration, reduces to $m + 1$ for the bipartite setup.

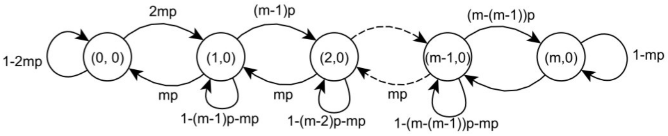


Figure 5. Graph representation for the bipartite ($n = 2$) star graph assuming that the success probability p of filling a memory cell is small so that it only appears in first order and at most one memory cell can be filled in each storage step. The vertices $(k_1^\downarrow, k_2^\downarrow)$ are occupation numbers of parties' memory cells in decreasing order. The last number is always zero because as soon as the corresponding party also fills one memory cell, a GHZ measurement is immediately performed, which empties this memory again.

It is worthwhile to note that the transition probability from $(k_1^\downarrow, k_2^\downarrow) = (0, 0)$ to $(k_1^\downarrow, k_2^\downarrow) = (1, 0)$ is $2mp$ rather than mp because any of the two parties can fill one of its memories. In other words, the ordered configuration $(k_1^\downarrow, k_2^\downarrow) = (1, 0)$ corresponds to the two unordered configurations $(k_1, k_2) = (1, 0)$ and $(k_1, k_2) = (0, 1)$, and the transition probability from $(0, 0)$ to each of these configurations is mp. However, the transition probability from $(k_1^\downarrow, k_2^\downarrow) = (1, 0)$ to $(k_1^\downarrow, k_2^\downarrow) = (2, 0)$ is $(m-1)p$, because now the order of the parties is fixed.

Recall that a Markov chain can be represented as a directed graph, where the vertices V are configurations and the edges E are possible transitions between them in one time step (with nonzero probabilities). In our case, $V = \{0, \ldots, m\}^n$. Then, the stationary distribution in the asymptotic limit can be obtained via the Markov chain tree theorem:

Theorem 1 ([21]). *Let a stochastic matrix T define an irreducible finite Markov chain with stationary distribution $\pi^* = (\pi^*)_{\mathbf{k}\in V}$. Then*

$$\pi^*_{\mathbf{k}} = \frac{\|\mathcal{A}_{\mathbf{k}}\|}{\|\mathcal{A}\|}. \tag{20}$$

Here, $\mathcal{A}_{\mathbf{k}}$ denotes the set of arborescences of a chosen root $\mathbf{k}$, and $\mathcal{A}$ is the set of all arborescences of the graph G. An arborescence with root $\mathbf{k}$ is a set of edges $A \subseteq E$, that fulfills the following properties:

- each vertex $\mathbf{k}' \in V$ has a directed path in the subgraph $G' = (V, A)$ to the chosen root $\mathbf{k}$.
- each vertex, except the root $\mathbf{k}$, has precisely one outgoing edge.

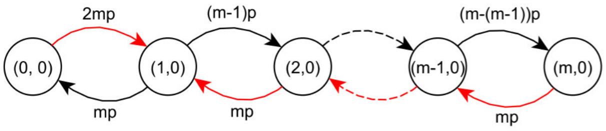


Figure 6. The only arborescence $A_{(1,0)}$ for the second node (root) $(1,0)$, i.e., the set of edges that lead to a directed path from all nodes to the chosen root $(1,0)$, is shown in red. Since self-loops do not contribute to the calculation, they are not shown in the graph representation here.

- the root $\mathbf{k}$ has no outgoing edge.

Each $\pi_{\mathbf{k}}$ is the probability of finding configuration $\mathbf{k}$ in the multipartite quantum repeater. In the graphical representation, each vertex $\mathbf{k}$ represents one such configuration. The weight $\|A\|$ of an arborescence A is given by the product of the weights of the edges p_e (in our case, probabilities of the corresponding transitions) included in that arborescence:

$$\|A\| = \prod_{e\in A} p_e. \tag{21}$$

Then, $\|\mathcal{A}_{\mathbf{k}}\|$ is the sum over the weights of all arborescences within the set $\mathcal{A}_{\mathbf{k}}$ and $\|\mathcal{A}\|$ is the sum over the weights of all arborescences of the graph G:

$$\begin{aligned}\|\mathcal{A}_{\mathbf{k}}\| &= \sum_{A\in\mathcal{A}_{\mathbf{k}}} \|A\|,\\ \|\mathcal{A}\| &= \sum_{A\in\mathcal{A}} \|A\| = \sum_{\mathbf{k}\in V} \|\mathcal{A}_{\mathbf{k}}\|.\end{aligned} \tag{22}$$

Thus, to calculate $\pi'_{\mathbf{k}}$ for all roots $\mathbf{k}$ (i.e., all configurations), it is required to find all arborescences in the graph and calculate their weights. The Markov chain tree theorem can also be applied to graphs with self-loops, which, however, do not contribute to arborescences and can thus be ignored [21].

We find that, in our case, each root has a single arborescence only: Starting from each end of the chain, all edges pointing towards the root are part of the arborescence. An example for the arborescence $A_{(1,0)}$ of root $(1,0)$ is shown in Fig. 6. By applying the Markov chain tree theorem to the bipartite setup, we find for the weights $W_{\mathbf{k}}$ for each root $\mathbf{k} \in \{(0,0),(1,0),\ldots,(m,0)\}$ the following expressions:

$$\|\mathcal{A}_{(0,0)}\| = (mp)^m, \tag{23}$$

$$\|\mathcal{A}_{(1,0)}\| = 2(mp)^m, \tag{24}$$

$$\|\mathcal{A}_{(k,0)}\| = 2m^{m-k+1}p^m\frac{(m-1)!}{(m-k)!} \quad \text{for} \quad 1\le k\le m. \tag{25}$$

To calculate the average number of GHZ measurements $\langle L\rangle$ per round, we use Eq. (19). Due to the assumption that at most one memory can be filled per round, maximally one GHZ measurement can be performed. The stationary distribution is calculated with Eq. (3). We see from the graph that only the configuration $\mathbf{k} = (0,0)$ at the beginning of a round leads to no GHZ measurement since, in our approximation, only one memory can be filled in one round. All other configurations $\mathbf{k} \in \{(1,0),(2,0),\ldots,(m,0)\}$ lead to a GHZ measurement with the probability pm of increasing k_1 from 0 to 1. For the average number of GHZ measurements, we then find:

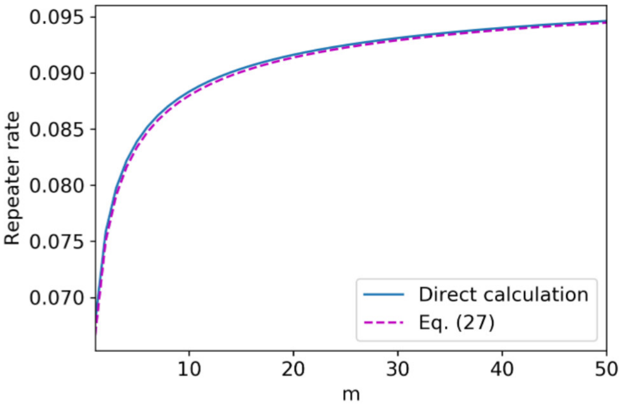


Figure 7. Repeater rate $R_\infty = \langle L\rangle/m$ for a bipartite network for different number of memories per party m. The success probability is chosen to be $p = 0.1$. The analytical approximation is calculated via Eq. (27), while the direct calculation is obtained via $T^n\pi_{init}$ for large n.

$$\begin{aligned}\langle L\rangle &= pm\left(\frac{\sum_{k'=1}^m \|\mathcal{A}_{(k',0)}\|}{\sum_{k=0}^m \|\mathcal{A}_{(k,0)}\|}\right)\\ &= pm\left(1-\frac{\|\mathcal{A}_{(0,0)}\|}{\sum_{k=0}^m \|\mathcal{A}_{(k,0)}\|}\right)\\ &= pm\left(1-\frac{1}{1+2\sum_{k=1}^m m^{-k+1}\frac{(m-1)!}{(m-k)!}}\right).\end{aligned} \tag{26}$$

For the repeater rate, it then holds:

$$R_\infty = p\left(1-\frac{1}{1+2\sum_{k=1}^m m^{-k+1}\frac{(m-1)!}{(m-k)!}}\right) \tag{27}$$

We can find asymptotic behavior for large m as follows:

$$\begin{aligned}\sum_{k=1}^m m^{-k+1}\frac{(m-1)!}{(m-k)!} &= \frac{m!}{m^m}\sum_{k=0}^{m-1}\frac{m^k}{k!}\\ &\approx \sqrt{2\pi m}\,e^{-m}\sum_{k=0}^{m-1}\frac{m^k}{k!} = \sqrt{2\pi m}\,F_m(m-1),\end{aligned} \tag{28}$$

where F_m is the cumulative distribution function of the Poisson probability distribution with the expectation m.

Due to the central limit theorem, the Poisson distribution is approximated by the normal distribution for large m, which is symmetric with respect to the expectation value. Hence, $F_m(m-1) \approx 1/2$ for large m (actually, already for $m = 5$ the approximation is good). We obtain then

$$R_\infty \approx p\left(1 - \frac{1}{1+\sqrt{2\pi m}}\right). \tag{29}$$

In Fig. 7, we compare the repeater rate for the bipartite network with up to $m = 50$ memories per party and a success probability of $p = 0.1$ determined in two different ways. We compare it with a direct calculation of $T^n \pi_{init}$ for large n. In the approximate solution, we use Eq. (27) to calculate the repeater rate. Fig. 7 shows that the approach of small p leads to a lower bound that provides good results also for larger m.

C. Larger network sizes

For networks with $n > 2$ parties and memory multiplexing $m \geq 1$, it is hard to find a generalization by proceeding analogously to the bipartite network, even under the assumption that $p \ll 1$. This is because the resulting graphs representing the Markov chains have more than one arborescence per root i. This holds already in the simplest case with $n = 3$ and $m = 2$. Here, each root has three arborescences. The number of arborescences can be determined using Tutte's directed matrix-tree theorem:

Theorem 2 ([22]). *Let $G = (V, E)$ be a directed graph and $\mathcal{L}$ a matrix with entries*

$$L_{\mathbf{k},\mathbf{k}'} = \begin{cases} \deg_{in}(\mathbf{k}'), & \text{if } \mathbf{k}' = \mathbf{k}, \\ -1, & \text{if } \mathbf{k} \neq \mathbf{k}' \text{ and } (\mathbf{k},\mathbf{k}') \in E, \\ 0 & \text{otherwise,} \end{cases} \tag{30}$$

$\mathbf{k}, \mathbf{k}' \in V$, where $\deg_{in}(\mathbf{k}')$ is the in-degree of vertex $\mathbf{k}'$. The number of arborescences $N_\mathbf{k}$ with root $\mathbf{k} \in V$ is then given by

$$N_\mathbf{k} = \det\left(\hat{\mathcal{L}}_\mathbf{k}\right) \tag{31}$$

where $\hat{\mathcal{L}}_\mathbf{k}$ is the matrix produced by deleting the $\mathbf{k}$-th row and column from $\mathcal{L}$.

Fig. 8 shows the increase of $N_\mathbf{k}$ up to $m = 10$ memories for a tripartite network for the first reduction of the configuration space described in Sec. IV A, i.e., the configuration is given by the (unordered) occupation numbers of three parties (k_1, k_2, k_3), under the approximation $p \ll 1$ (again, p is taken into account only up to the first order). It can be seen that even under the assumption of small p, it is not possible to find a generalized expression for $\langle L \rangle$.

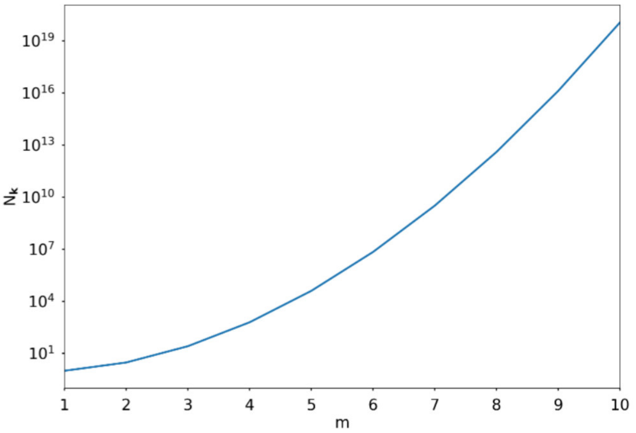


Figure 8. Number of arborescences $N_\mathbf{k}$ for root $\mathbf{k} = (m, m, 0)$ for a tripartite network with up to ten memories. Tutte's directed matrix-tree theorem (i.e., Eq.(31)) is used for the calculation.

Nevertheless, we can construct bounds for the average number of GHZ measurements for the case of n parties each having m memories. Suppose we treat the memories independently of each other (i.e., no multiplexing is performed despite the multimemory setup). This case leads to the following lower bound:

$$\langle L \rangle \geq m \langle L_1 \rangle \tag{32}$$

An upper bound for $\langle L \rangle$ can be achieved from numerical simulations performed analogously to Ref. [8]:

$$\langle L \rangle \leq pm. \tag{33}$$

One can understand it as follows. After the measurement step, at least one party has no filled memories. The average number of memories filled in the next storage step for this party is pm. Since L cannot be larger than the minimal number of filled memories among the parties, $\langle L \rangle$ cannot be larger than pm.

However, comparison with simulation show that both bounds (32) and (33) are loose. In the following subsection, we derive an approximate formula for the general repeater rate, making different approximations.

D. Probabilistic model of the multipartite quantum repeater and the repeater rate for the general case

The purpose of this subsection is to analyze the general case of an arbitrary number of parties n and the number of memories per party m. Up to now, we have rigorous bounds only for the cases $m = 1$ and arbitrary n (Eqs. (13) and (14)) and $n = 2$ and arbitrary m (Eq. (26)). As we saw above, it is hard to generalize Eq. (26) to the case of an arbitrary n. We need a formula from which the asymptotic behavior with respect to n and m could be easily derived.

Our derivation of such a formula will be based on the following probabilistic model of the multipartite quantum repeater:

- $X_{i,k}$ is the random variable of the number of filled memories (from 0 to m) for the ith party ($i = 1,\dots,n$) after the round k, i.e., before the $(k+1)$th storage stage, $t = 0, 1, 2, \dots$ The initial number of filled memories is $X_{i,0} = 0$.

- $Y_{i,k}$ is the number of new filled memories for the party i in the storage stage in the round k. It takes values from 0 to $m - X_{i,k}$ and obeys the binomial distribution:

$$\Pr[Y_{i,k} = y | X_{i,k-1} = x] = \binom{m-x}{y} p^y (1-p)^{m-x-y}, \tag{34}$$

$y = 0, 1, \dots, m - x$.

- $Z_{i,k} = X_{i,k-1} + Y_{i,k}$ is the number of the filled memories for the party i before the measurement in the round k.

- $L_k = \min_i Z_{i,k}$ is the number of GHZ measurements in the round k.

- Measurement stage:

$$X_{i,k} = Z_{i,k} - L_k. \tag{35}$$

- $\langle L\rangle = \lim_{k\to\infty}\langle L_k\rangle$ is the long-term expectation value of L_k, which we want to find (or approximate). If the probability distribution of L_k converges to a limiting distribution, L can be considered a random variable with this distribution.

However, the probabilistic analysis of the model described above is problematic due to dependencies of $X_{i,k}$ on each other by means of the subtraction of L_k, which depend on $X_{i,k-1}$ for all i. To break this dependency, we develop a simplified model. To do this, let us iterate Eq. (35):

$$X_{i,k} = X_{i,0} + \sum_{k'=1}^{k} Y_{i,k'} - \sum_{k'=1}^{k} L_{k'}. \tag{36}$$

The intuition related to the law of large numbers tells that, for large k, $\sum_{k'=1}^{k} L_{k'}$ can be replaced by $k\langle Y\rangle$. In other words, random $L_{k'}$ can be replaced by a fixed number.

In view of this intuition, consider the random variables $\tilde{X}_{i,k}$, $\tilde{Y}_{i,k}$, $\tilde{Z}_{i,k}$, and $\tilde{L}_k$ with the following modification: the recurrence equation (35) is replaced by

$$\tilde{X}_{i,k} = \tilde{Z}_{i,k} - l \tag{37}$$

for a *fixed* number l to be determined. Subsequently, the random variables $\tilde{X}_{i,k}$ can take negative values. Namely, if $\tilde{X}_{i,0} = 0$, then $X_{i,k}$ can take arbitrary integer values from $-lk$ (in the case $Y_{i,k'} = 0$, $k = 1,\dots,k$) to m. Then, $Y_{i,k}$ also can take values larger than m (if $X_{i,k-1}$ is negative). Formula (34) holds with X and Y replaced by $\tilde{X}$ and $\tilde{Y}$.

In this case, $X_{i,k}$ and $X_{j,k}$ are independent for $i \neq j$ (and identically distributed because they obey the same recurrence relation (37)). In other words, the dynamics of the memory occupation numbers for different participants are uncoupled, which largely simplifies the analysis.

In the original model all $X_{i,k}$ are nonnegative or, equivalently,

$$\min(X_{1,k},\dots,X_{n,k}) \geq 0. \tag{38}$$

In the simplified model, we replace this requirement with the following one: The number l is chosen as a maximal value such that the inequality

$$\lim_{k\to\infty}\langle \min(\tilde{X}_{1,k},\dots,\tilde{X}_{n,k})\rangle \geq 0 \tag{39}$$

holds, i.e., the long-time average of the minimum of the occupation numbers is non-negative. The intuition behind such a replacement is again the law of large numbers: we can hope that, in the asymptotic case, the fluctuations around the expectation value are negligible.

One can suggest the following financial analogy. If $\tilde{X}_{i,k}$ is the current "wealth" of the ith party, we take a fixed "tax" l and allow the party to "borrow" money for some time (negative $\tilde{X}_{i,k}$), but, on average, the minimal "wealth" among the parties must remain non-negative.

Then, under the additional assumption of normal distribution for $X_{i,k}$ and in the asymptotic case of large n, one can derive the following formula for the maximal l, which is an estimate for $\langle L\rangle$ in the original model (see Appendix B):

$$\langle L\rangle \approx pm\left(\sqrt{\frac{\alpha^2\beta}{4m}\ln n + 1} - \sqrt{\frac{\alpha^2\beta}{4m}\ln n}\right)^2. \tag{40}$$

Here $\beta = (1-p)/(2-p)$,

$$\alpha = \sqrt{2}\left(1 - \frac{\ln(4\pi\ln n) - 2\gamma}{4\ln n}\right) \tag{41}$$

and γ is the Euler-Mascheroni constant.

Note that for $n \to \infty$, we get

$$\langle L\rangle \approx \frac{pm^2}{\alpha^2\beta\ln n}, \tag{42}$$

i.e., logarithmical decrease, which agrees with the results for the case of no multiplexing (Sec. III). The quadratic dependence on m does not contradict the upper bound (33) with the linear dependence because, for large n, the right-hand side of Eq. (42) is smaller than pm. The quadratic increase with m can be understood by the double role of memory multiplexing. First, it increases the average number of new Bell links per round. E.g., if all memories of a party are empty, then pm is the average

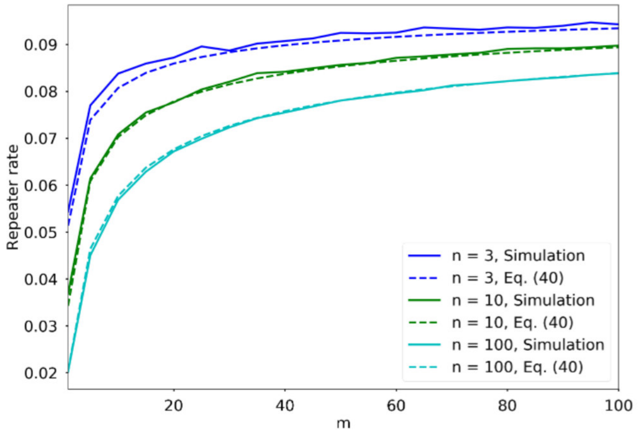


Figure 9. Comparison between simulations and the approximate Eq. (40) for the repeater rate $R_\infty = \langle L \rangle / m$ with the probability of successfully storing entangled qubits in memory is $p = 0.1$. The repeater rate is shown for varying numbers of parties n and memories m.

number of filled memories after a storage step. Second, if a party already has one filled memory and waits for other parties, multiple memories allows him not to waste time, but to establish additional Bell links for future measurements, so, after the measurement step, the parties start not from scratch, but already have Bell links.

Interestingly, though strictly speaking, formula (40) uses the approximation of large n, substitution of $n = 1$ in Eq. (40) gives the exact result for this case $\langle L \rangle = pm$.

Also we see that, for large m, the repeater rate (11) saturates:

$$R_\infty = \frac{\langle L \rangle}{m} \approx p \left(1 - \sqrt{\frac{\alpha^2 \beta}{4m} \ln n} \right), \tag{43}$$

so the effect of an additional memory for each party does not increase R_∞. Also we see the agreement with the large m asymptotics (29) for the case $n = 2$: The repeater rate saturates as $c/\sqrt{m}$ with some constant c. Eqs. (29) and (43) (for small p) give close values of c.

To justify the approximation, we show the repeater rate for varying network sizes in Fig. 9 comparing the approximation formula obtained in Eq. (40) and the simulation of the multipartite quantum repeater performed. Eq. (40) gives a very good approximation for the whole range of parameters (though formally it was derived for the asymptotic case of large n).

In Fig. 10, we give an overview of the scaling of the repeater rate calculated via the approximation formula up to a network size of $n = 150$ parties, each having up to $m = 100$ memories. The plot shows that, for small m, the repeater rate grows linearly with m, but then the growth slows down. The plot confirms the saturation of the repeater rate (43). The red dots indicate the limit at

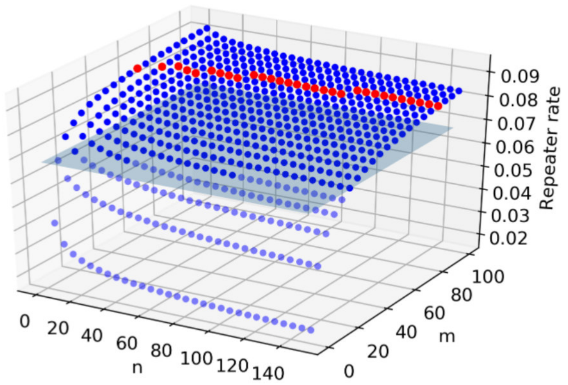


Figure 10. Repeater rate $R_\infty = \langle L \rangle / m$ obtained by formula (40) for different network sizes depending on n and m and a success probability of $p = 0.1$. For a rate beyond the line of red dots, the rate does not further increase with the number of memories. The light blue plane shows the minimal achievable repeater rate in a bipartite setup with a single link for both parties.

which adding more memories to the network no longer significantly increases the repeater rate, i.e., the difference in the repeater rate becomes smaller than 0.0001 with increasing m. For networks with more than 100 parties, this limit is at about $m = 85$ per party.

The light blue plane shown in Fig. 10 gives the repeater rate achieved in a bipartite network with only one memory per party ($R_\infty = 0.068$). For all network sizes considered here, this threshold can be reached by increasing the number of memories per party. Thus, one can achieve the repeater rate of a bipartite network also in a multipartite network.

V. CONCLUSION

Establishing entangled states between end nodes in large quantum networks is one of the main challenges to allowing the end nodes to communicate or share secret keys. To enlarge the distance between the end nodes and overcome distance-based fiber losses, quantum repeaters are used. In our work, we have analyzed the repeater rate of such multipartite quantum repeaters (multipartite quantum repeaters), i.e., the average number of distributed GHZ states per round in the stationary (long-term) regime. We have considered both single links between the end nodes and the central multipartite quantum repeater and multiple links (multiplexing). Our results can be used to plan quantum networks and estimate the achievable repeater rates for various network sizes. The optimal number of memories can be calculated, especially when the number of parties increases.

For the single link case, we have derived Eqs. (13) and

(14), which coincide with the rate of a chain of repeaters between two participants obtained in Refs. [9, 17]. For the multiplexing setup and two parties, we have derived an approximate formula (27) for the repeater rate based on the approximation of small success probabilities p.

Finally, we have derived an approximate formula to calculate the average number of GHZ measurements per round for an arbitrary number of participants and memories. This approximation gives very good agreement with the simulation.

It turns out that the repeater rate saturates for a large number of memories, i.e., the number of GHZ measurements per round grows not faster than linearly with the number of memories per party. Additionally, we observe that as the number of participants n increases, the repeater rate decreases slowly as $(\ln n)^{-1}$, provided the number of memories per party remains constant.

In our work, we show that there is a maximal number of memories that helps to increase the repeater rate. Adding more memories does not lead to higher rates. In future work, integrating entanglement purification will be of great interest [23]. It will be interesting to analyze how the fidelities change due to the underlying network structure and how entanglement purification can increase the fidelity [24]. It should be investigated for which network sizes multiplexing leads to higher fidelities than entanglement purification. In a further step, applications such as conference key agreement [16] should also be included in the analysis.

VI. ACKNOWLEDGMENT

We thank Jörg Rothe and Moritz Wiese for valuable discussions. This work was funded by the Federal Ministry of Education and Research BMBF (QR.X with Project No. 16K1SQ023 and QuKuK with Project No. 16KIS1618K).

[1] V. Scarani, H. Bechmann-Pasquinucci, N. J. Cerf, M. Dušek, N. Lütkenhaus, and M. Peev, The security of practical quantum key distribution, Rev. Mod. Phys. **81**, 1301 (2009).

[2] H.-J. Briegel, W. Dür, J. I. Cirac, and P. Zoller, Quantum repeaters: The role of imperfect local operations in quantum communication, Phys. Rev. Lett. **81**, 5932 (1998).

[3] W. Dür, H.-J. Briegel, J. I. Cirac, and P. Zoller, Quantum repeaters based on entanglement purification, Phys. Rev. A **59**, 169 (1999).

[4] L. Hartmann, B. Kraus, H.-J. Briegel, and W. Dür, Role of memory errors in quantum repeaters, Phys. Rev. A **75**, 032310 (2007).

[5] G. Avis, F. Rozpędek, and S. Wehner, Analysis of multipartite entanglement distribution using a central quantum-network node, Physical Review A **107**, 10.1103/physreva.107.012609 (2023).

[6] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, Multiplexed memory-insensitive quantum repeaters, Phys. Rev. Lett. **98**, 060502 (2007).

[7] S. Abruzzo, H. Kampermann, and D. Bruß, Finite-range multiplexing enhances quantum key distribution via quantum repeaters, Physical Review A **89**, 012303 (2014).

[8] J. A. Kunzelmann, H. Kampermann, and D. Bruß, Multipartite multiplexing strategies for quantum routers, Phys. Rev. A **110**, 032617 (2024).

[9] E. Shchukin, F. Schmidt, and P. van Loock, On the waiting time in quantum repeaters with probabilistic entanglement swapping, Phys. Rev. A **100**, 032322 (2019).

[10] S. Khatri, C. T. Matyas, A. U. Siddiqui, and J. P. Dowling, Practical figures of merit and thresholds for entanglement distribution in quantum networks, Phys. Rev. Res. **1**, 023032 (2019).

[11] S. E. Vinay and P. Kok, Statistical analysis of quantum-entangled-network generation, Phys. Rev. A **99**, 042313 (2019).

[12] L. Kamin, E. Shchukin, F. Schmidt, and P. van Loock, Exact rate analysis for quantum repeaters with imperfect memories and entanglement swapping as soon as possible, Phys. Rev. Res. **5**, 023086 (2023).

[13] O. A. Collins, S. D. Jenkins, A. Kuzmich, and T. A. B. Kennedy, Multiplexed memory-insensitive quantum repeaters, Phys. Rev. Lett. **98**, 060502 (2007).

[14] L. Jiang, J. M. Taylor, and M. D. Lukin, Fast and robust approach to long-distance quantum communication with atomic ensembles, Phys. Rev. A **76**, 012301 (2007).

[15] M. Razavi, M. Piani, and N. Lütkenhaus, Quantum repeaters with imperfect memories: Cost and scalability, Phys. Rev. A **80**, 032301 (2009).

[16] G. Murta, F. Grasselli, H. Kampermann, and D. Bruß, Quantum conference key agreement: A review, Advanced Quantum Technologies **3**, 2000025 (2020).

[17] N. K. Bernardes, L. Praxmeyer, and P. van Loock, Rate analysis for a hybrid quantum repeater, Phys. Rev. A **83**, 012323 (2011).

[18] W. Szpankowski and V. Rego, Yet another application of a binomial recurrence order statistics, Computing **43**, 401 (1990).

[19] L. Praxmeyer, Reposition time in probabilistic imperfect memories (2013), arXiv:1309.3407 [quant-ph].

[20] B. Eisenberg, On the expectation of the maximum of iid geometric random variables, Stat. Probabil. Lett. **78**, 135 (2008).

[21] F. Leighton and R. Rivest, Estimating a probability using finite memory, IEEE Transactions on Information Theory **32**, 733 (1986).

[22] W. T. Tutte, *Graph theory*, Vol. 21 (Cambridge university press, 2001).

[23] C. H. Bennett, G. Brassard, S. Popescu, B. Schumacher, J. A. Smolin, and W. K. Wootters, Purification of noisy entanglement and faithful teleportation via noisy channels, Physical Review Letters **76**, 722–725 (1996).

[24] B. Davies, Á. G. Iñesta, and S. Wehner, Entanglement buffering with two quantum memories, Quantum **8**, 1458

(2024).
[25] H. A. David and H. N. Nagaraja, *Order Statistics* (John Wiley & Sons, 2003).

Appendix A: Derivation and proof of the asymptotic repeater rate without multiplexing

Here, we provide our derivation of the repeater rate for the setup with a single memory per party only. We start with the transition matrix T' which is given by

$$T' = \sigma\mu = \left(\sigma^{(1)}\right)^{\otimes N} \mu$$

Note that for this derivation, we consider the transition matrix in reverse order. This can be done since the stationary distribution is independent of the initial distribution $\pi^{(0)}$. This transition matrix already provides the distribution after the process of storing qubits in memory. It is, therefore, not necessary to apply another map. This order is chosen here because it simplifies the following calculations.

The storage process for one party with a single memory is given by

$$\sigma^{(1)} = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix} \tag{A1}$$

with p being the success probability of the optical fiber. The total storage map for n parties follows from the tensor product:

$$\sigma = \begin{pmatrix} 1-p & 0 \\ p & 1 \end{pmatrix}^{\otimes n} \tag{A2}$$

A measurement is performed only when the memory of every party is filled. In that case, the memories are emptied; in all other cases, the memory configuration does not change. Therefore, the measurement map is given by

$$\mu = \mathbb{1}_{2^n} - \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes n} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes n} \tag{A3}$$

By combining both maps, we find the following transition map:

$$\begin{aligned} T' &= \sigma + \sigma\left(-\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes n} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes n}\right) \\ &\equiv \sigma + \sigma X_n \end{aligned} \tag{A4}$$

where we set $\left(-\begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}^{\otimes n} + \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix}^{\otimes n}\right) = X_n$. In the next step, we calculate T'^s to get the stationary distribution of the Markov chain $\pi^* = \lim_{s\to\infty} T'^s \pi^{(0)}$. It is possible to rewrite T'^s by

$$T'^s = \sigma^s + \sum_{i=1}^{s} A_i X_n B_{s-i} \tag{A5}$$

with the second term given by

$$B_{s-i} = \sigma^{s-i}. \tag{A6}$$

The first term is of the following form:

$$A_i = \sum_{j=1}^{i} c_j^{(i)} \sigma^j. \tag{A7}$$

The prefactors $c_j^{(i)}$ are given by

$$c_j^{(i)} = \begin{cases} \sum_{l=1}^{i-1} c_1^{(l)} \left(\wp_{i-l} - 1\right), & j = 1 \\ c_{j-1}^{(i-1)}, & j > 1 \end{cases} \tag{A8}$$

where the first factor is fixed to $c_1^{(1)} = 1$ and with $\wp_s = (1-(1-p)^s)^n$.

Before going on, we explain Eq. (A5) in detail and prove its validity via induction. To understand where the decomposition of T'^s comes from, it is best to consider a concrete example (e.g., T'^2) first. We note that with

$$\sigma^s = \begin{pmatrix} (1-p)^s & 0 \\ 1-(1-p)^s & 1 \end{pmatrix} \tag{A9}$$

we find the following:

$$\begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes n} \sigma \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes n} = \begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes n} \sigma^s \begin{pmatrix} 0 \\ 1 \end{pmatrix}^{\otimes n} = 1 \tag{A10}$$

and

$$\begin{pmatrix} 0 & 1 \end{pmatrix}^{\otimes n} \sigma^s \begin{pmatrix} 1 \\ 0 \end{pmatrix}^{\otimes n} = (1-(1-p)^s)^n \tag{A11}$$

Therefore, we can rewrite the term $X_n \sigma^s X_n$ by the term $(\wp_s - 1)\, X_n$ with $\wp_s = (1-(1-p)^s)^n$, which only depends on the number of parties n and the number of rounds s. T'^2 can thus be reformulated in the following way:

$$\begin{aligned} T'^2 &= (\sigma + \sigma X_n)(\sigma + \sigma X_n) \\ &= \sigma^2 + \sigma X_n \sigma + \left((\wp_1 - 1)\,\sigma + \sigma^2\right) X_n \sigma^0 \end{aligned} \tag{A12}$$

Here, we already find the structure from Eq. (A5). To see that this is not by chance, we additionally calculate T'^3 in a similar way:

$$\begin{aligned} T'^3 &= \left(\sigma^2 + \sigma X_n \sigma + (\wp_1 - 1)\,\sigma X_n + \sigma^2 X_n\right)(\sigma + \sigma X_n) \\ &= \sigma^3 + \sigma X_n \sigma^2 + \left((\wp_1 - 1)\,\sigma + (\sigma)^2\right) X_n \sigma \\ &+ \left\{\left((\wp_2 - 1) + (\wp_1 - 1)^2\right)\sigma + (\wp_1 - 1)\,\sigma^2 + \sigma^3\right\} X_N \sigma^0. \end{aligned} \tag{A13}$$

This leads again to an expression of the form:

$$T'^s = \sigma^s \sum A_i X_n B_{s-i} \tag{A14}$$

with $B_{s-i} = \sigma^{s-i}$. Now, we can guess the general behavior for a term $T'^s = T'^{s-1} T'$:

- Multiplying T'^{s-1} with σ leaves the factor A_i untouched and only increases the power of the last term by one.
- Multiplying T'^{s-1} with σX_n creates a new term in the sum since i increases by one, i.e., there is a new term with factors A_s and B_0.
- The new factor A_s for the last term is given by

$$A_s = \sum_{j=1}^{s-1} A_j \left(\wp_{s-j} - 1\right) + \sigma^n \tag{A15}$$

By rewriting the term for the A_i to the form

$$A_i = \sum_{j=1}^{i} = c_j^{(i)} \sigma^j, \tag{A16}$$

one finds the prefactors $c_j^{(i)}$ as given in Eq. (A8):

$$c_j^{(i)} = \begin{cases} \sum_{l=1}^{i-1} c_1^{(l)} \left(\wp_{i-l} - 1\right), & j = 1 \\ c_{j-1}^{(i-1)}, & j > 1 \end{cases} \tag{A17}$$

with the first factor $c_1^{(1)} = 1$.

Let us prove the equations

$$T'^s = \sigma^s + \sum_{i=1}^{s} A_i X_n B_{s-i}, \tag{A18}$$

$$A_i = \sum_{j=1}^{i-1} A_j \left(\wp_{i-j} - 1\right) + \sigma^i, \tag{A19}$$

$$B_i = \sigma^i \tag{A20}$$

by induction.

Proof. Starting with the base case, we show equality for $s = 2$. From Eq. (A12), it follows

$$\begin{aligned} T'^2 &= \sigma^2 + \sigma X_n \sigma + \left(\wp_1 - 1\right) \sigma X_n + \sigma^2 X_n \\ &= \sigma^2 + \sigma X_n \sigma + \left\{\left(\wp_1 - 1\right)\sigma + \sigma^2\right\} X_n \\ &= \sigma^2 + A_1 X_n B_1 + A_2 X_n B_0 \\ &= \sigma^s + \sum_{i=1}^{s} A_i X_n B_{s-i} \end{aligned}$$

with A_i given in Eq. (A19) and B_i given in Eq. (A20):

$$\begin{aligned} A_1 &= \sigma, \quad A_2 = \left(\wp_1 - 1\right)\sigma + \sigma^2, \\ B_1 &= \sigma, \quad B_0 = 1. \end{aligned}$$

In the general case, we assume that the equations (A18)-(A20) hold for any $s \in \mathbb{N}$. In the induction step, we show that if the statement holds for any $s \in \mathbb{N}$, then $s + 1$ follows:

$$\begin{aligned} T'^{s+1} &= (T'^s)T' \\ &= \left(\sigma^s + \sum_{i=1}^{s} A_i X_n B_{s-i}\right) \left(\sigma + \sigma X_n\right) \\ &= \sigma^{s+1} + \sigma^{s+1} X_n + \sum_{i=1}^{s} A_i X_n B_{s-i+1} \\ &\qquad + \sum_{i=1}^{s} A_i \left(\wp_{s-i+1} - 1\right) X_n \\ &= \sigma^{s+1} + \sum_{i=1}^{s} A_i X_n B_{s-i+1} \\ &\qquad + \left(\sum_{i=1}^{s} A_i \left(\wp_{s-i+1} - 1\right) + \sigma^{s+1}\right) X_n \\ &= \sigma^{s+1} + \sum_{i=1}^{s} A_i X_n B_{s-i+1} + A_{s+1} X_n B_0 \\ &= \sigma^{s+1} + \sum_{i=1}^{s+1} A_i X_n B_{s-i+1} \\ &= T'^{s+1}. \end{aligned}$$

□

To get the probability distribution in the asymptotic limit, we choose an initial configuration and further calculate $T'^{s\to\infty}\pi^{(0)}$. It turns out that many terms cancel when choosing $\pi^{(0)} = \begin{pmatrix}0 & 1\end{pmatrix}^T$, which means that in the initial configuration, all memories are filled. As the goal is to calculate the average number of GHZ measurements per round, we are only interested in the last entry of the stationary distribution π^*, which gives the probability that all memories are filled (see Eq. (12)). Therefore, we need to calculate

$$\begin{aligned} &\begin{pmatrix}0 & 1\end{pmatrix}^{\otimes n} T'^s \begin{pmatrix}0 \\ 1\end{pmatrix}^{\otimes n} \\ &= \begin{pmatrix}0 & 1\end{pmatrix}^{\otimes n} \left[\sigma^s + \sum_{i=1}^{s} A_i X_n B_{s-i}\right] \begin{pmatrix}0 \\ 1\end{pmatrix}^{\otimes n} \\ &= 1 + \sum_{i=2}^{s} \sum_{j=1}^{i-1} c_1^{(j)} \left(\wp_{i-j} - 1\right) \\ &= \sum_{j=1}^{s} c_1^{(j)} \end{aligned} \tag{A21}$$

By rearranging the sums and considering $\lim_{s\to\infty}$, the asymptotic result follows:

$$\begin{aligned}(0\ \ 1)^{\otimes n}\, T'^s \begin{pmatrix}0\\1\end{pmatrix}^{\otimes n} &= \sum_{j=1}^{\infty} c_1^{(j)}\\ &= 1 + \sum_{j=1}^{\infty} c_1^{(j)} \sum_{i=1}^{\infty} (\wp_i - 1)\end{aligned} \tag{A22}$$

$$\begin{aligned}\Rightarrow (0\ \ 1)^{\otimes n}\, T'^s \begin{pmatrix}0\\1\end{pmatrix}^{\otimes n} &= \frac{1}{1+\sum_{j=1}^{\infty}(1-\wp_j)}\\ &= \frac{1}{1+\sum_{j=1}^{\infty}\left(1-\left(1-(1-p)^j\right)^n\right)}\\ &= \langle L_1\rangle\end{aligned} \tag{A23}$$

with the sum over the rounds, here denoted as j.

Appendix B: Stationary expectation and dispersion for the simplified model

The aim of this section is to derive Eq. (40). Recall that $X_{i,k}$ are identically distributed. Denote $\langle X_{i,k}\rangle = \mu_k$ and $\operatorname{Var}[X_{i,k}] = \sigma_k^2$.

Lemma 1. *The following recurrence relations hold:*

$$\begin{aligned}\mu_{k+1} &= pm + (1-p)\mu_k - l,\\ \sigma_{k+1}^2 &= (1-p)^2\sigma_k^2 + p(1-p)(m-\mu_k).\end{aligned} \tag{B1}$$

Note that, unlike other steps of the present analysis, this lemma is a mathematically rigorous statement about a well-defined probabilistic model. The proof will be given later.

Take the limit $k \to \infty$ in Eqs. (B1) and denote $\mu = \lim_{k\to\infty}\mu_k$ and $\sigma^2 = \lim_{k\to\infty}\sigma_k^2$. We have

$$\begin{aligned}\mu &= m - \frac{l}{p},\\ \sigma^2 &= \frac{1-p}{2-p}\frac{l}{p}.\end{aligned} \tag{B2}$$

In order to calculate $\langle \min_i \tilde{X}_{i,k}\rangle$, see Ineq. (39), we need to know the distribution of $\tilde{X}_{i,k}$, not just the expectation value and variance. Numerical simulations show that, for large k, even for small n and m it is well approximated by the normal distribution with the expectation μ_k and the dispersion σ_k^2 given above. Under this assumption, we can use the known approximation (asymptotics) of the minimum of n identically normally distributed random variables as $n \to \infty$ [25]:

$$\langle \min(\tilde{X}_{1,k}, \ldots, \tilde{X}_{n,k})\rangle = \mu_k - \alpha\sigma_k\sqrt{\ln n}, \tag{B3}$$

and, thus,

$$\lim_{k\to\infty} \langle \min(\tilde{X}_{1,k}, \ldots, \tilde{X}_{n,k})\rangle = \mu - \alpha\sigma\sqrt{\ln n}, \tag{B4}$$

where

$$\alpha = \sqrt{2}\left(1 - \frac{\ln(4\pi\ln n) - 2\gamma}{4\ln n}\right) \tag{B5}$$

and γ is the Euler-Mascheroni constant. However, the simulations show that $\alpha = 1$ gives a good approximation for a broad range of parameters.

In order to find the maximal value $l_{\max}$ of l such that Ineq. (39) holds, we substitute Eqs. (B2) into Eq. (B4). This gives a quadratic function of $\sqrt{l}$. Its substitution into Ineq. (39) gives

$$l_{\max} = pm\left(\sqrt{\frac{\alpha^2\beta}{4m}\ln n + 1} - \sqrt{\frac{\alpha^2\beta}{4m}\ln n}\right)^2, \tag{B6}$$

where $\beta = (1-p)/(2-p)$. We take this as an approximation for $\langle L\rangle$ and obtain (40).

Let us again summarize the approximations used in this simplified model for the derivation of approximation (40) for $\langle L\rangle$:

- Substitution of random ℓ in the original model by a fixed number $\tilde{\ell}$ associated with the average value $\langle \ell\rangle$ in the stationary regime.
- We allow the occupation numbers $\tilde{X}_i$ to be negative and demand only that their stationary average values are non-negative.
- The use of the normal distribution for stationary occupation numbers $\tilde{X}_i$.
- The use of the asymptotic expressions $n \to \infty$ for the case of finite or even small n.

The first two assumptions are based on the law of large numbers (or neglection of fluctuations) argument, while the third one – on the central limit theorem argument. Also, these assumptions are justified by simulations. However, it would be interesting to obtain mathematically rigorous justifications.

Proof of Lemma 1. We have [see Eq. (34)]

$$\Pr[\tilde{Y}_{i,k} = y | \tilde{X}_{i,k-1} = x] = \binom{m-x}{y} p^y (1-p)^{m-x-y}, \tag{B7}$$

$y = 0, 1, \ldots, m-x$. Denote

$$\begin{aligned}\mathbb{E}[\tilde{Y}_{i,k} | \tilde{X}_{i,k-1} = x] &= \sum_{y=0}^{m-x} y \Pr[\tilde{Y}_{i,k} = y | \tilde{X}_{i,k-1} = x]\\ &= p(m-x)\end{aligned} \tag{B8}$$

the conditional expectation of $\tilde{Y}_{i,k}$ on the condition that $\tilde{X}_{i,k}$ takes the value x (in the second line we used the expression for the expectation value for the binomial distribution) and

$$\mathbb{E}[\tilde{Y}_{i,k} | \tilde{X}_{i,k-1}] = p(m - \tilde{X}_{i,k-1}) \tag{B9}$$

the conditional expectation of $\tilde{Y}_{i,k}$ conditioned on the random variable $\tilde{X}_{i,k}$ (informally speaking, regarding $\tilde{X}$ is a fixed number). This is in accordance with the standard definition of the conditional expectation and $\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]$ is still a random variable. Then

$$\begin{aligned}\langle Y_{i,k}\rangle &= \sum_{x=-\infty}^{m} \Pr[\tilde{X}_{i,k-1}=x]\,\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}=x] \\ &= \big\langle \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big\rangle = p(m-\mu_{k-1}).\end{aligned} \tag{B10}$$

Generally,

$$\langle A\rangle = \big\langle \mathbb{E}[A|\tilde{X}_{i,k-1}]\big\rangle \tag{B11}$$

for an arbitrary random variable A, which is a general property of the conditional expectation and which we will use.

Since $\tilde{Z}_{i,k} = \tilde{X}_{i,k-1} + \tilde{Y}_{i,k}$ and $\tilde{X}_{i,k} = \tilde{Z}_{i,k} - l$,

$$\langle \tilde{Z}_{i,k}\rangle = \mu_{k-1} + (m-\mu_{k-1})p = mp + (1-p)\mu_{k-1} \tag{B12}$$

and

$$\mu_k = \mathbb{E}[\tilde{X}_{i,k}] = mp + (1-p)\mu_{k-1} - l. \tag{B13}$$

We have obtained the first formula in Eqs. (B1). The calculation for the dispersion is as follows:

$$\begin{aligned}\sigma_k^2 &\equiv \mathrm{Var}[\tilde{X}_{i,k}] = \mathrm{Var}[\tilde{Z}_{i,k}] \\ &= \mathrm{Var}[\tilde{X}_{i,k-1}] + \mathrm{Var}[\tilde{Y}_{i,k}] + 2\,\mathrm{cov}[\tilde{X}_{i,k-1}, \tilde{Y}_{i,k}] \\ &= \sigma^2 + \mathrm{Var}[\tilde{Y}_{i,k}] + 2\,\mathrm{cov}[\tilde{X}_{i,k-1}, \tilde{Y}_{i,k}],\end{aligned} \tag{B14}$$

where $\mathrm{cov}[\tilde{X}_{i,k-1}, \tilde{Y}_{i,k}]$ denotes the covariance of the two random variables. We have

$$\begin{aligned}&\mathrm{cov}[\tilde{X}_{i,k-1}, \tilde{Y}_{i,k}] \\ &= \big\langle \big(\tilde{X}_{i,k-1} - \langle\tilde{X}_{i,k-1}\rangle\big)\big(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle\big)\big\rangle \\ &= \big\langle \mathbb{E}\big[\big(\tilde{X}_{i,k-1} - \langle\tilde{X}_{i,k-1}\rangle\big)\big(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle\big)\big|\tilde{X}_{i,k-1}\big]\big\rangle \\ &= \big\langle \big(\tilde{X}_{i,k-1} - \langle\tilde{X}_{i,k-1}\rangle\big)\,\mathbb{E}\big[\big(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle\big)\big|\tilde{X}_{i,k-1}\big]\big\rangle \\ &= \big\langle \big(\tilde{X}_{i,k-1} - \langle\tilde{X}_{i,k-1}\rangle\big)\,p(\mu_{k-1} - \tilde{X}_{i,k-1})\big\rangle \\ &= -p\langle(\tilde{X}_{i,k-1} - \mu_{k-1})^2\rangle = -p\sigma_k^2.\end{aligned} \tag{B15}$$

Here we have used Eqs. (B9)–(B11).

Now let us calculate $\mathrm{Var}[\tilde{Y}]$:

$$\begin{aligned}\mathrm{Var}[\tilde{Y}_{i,k}] &= \big\langle(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle)^2\big\rangle \\ &= \big\langle \mathbb{E}\big[(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle)^2\big|\tilde{X}\big]\big\rangle,\end{aligned} \tag{B16}$$

where we have again used Eq. (B11). Express now

$$\begin{aligned}&(\tilde{Y}_{i,k} - \langle\tilde{Y}_{i,k}\rangle)^2 \\ &= \Big\{\big(\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big) + \big(\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle\big)\Big\}^2 \\ &= \big(\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big)^2 + \big(\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle\big)^2 \\ &+ 2\big(\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big)\big(\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle\big)\end{aligned} \tag{B17}$$

and notice

$$\mathbb{E}\{(\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}])^2|\tilde{X}_{i,k-1}\} = p(1-p)(m - \tilde{X}_{i,k-1}) \tag{B18}$$

(the variance of the binomial distribution),

$$\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle = p(\mu - \tilde{X}_{i,k-1}), \tag{B19}$$

and

$$\begin{aligned}&\mathbb{E}\big\{\big(\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big)\big(\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle\big)\big|\tilde{X}_{i,k-1}\big\} \\ &= \big(\mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}] - \langle\tilde{Y}_{i,k}\rangle\big)\,\mathbb{E}\big\{\tilde{Y}_{i,k} - \mathbb{E}[\tilde{Y}_{i,k}|\tilde{X}_{i,k-1}]\big|\tilde{X}_{i,k-1}\big\} \\ &= 0\end{aligned} \tag{B20}$$

(the last factor is zero). Hence,

$$\begin{aligned}\mathrm{Var}[\tilde{Y}_{i,k}] &= \big\langle p(1-p)\tilde{X}_{i,k-1} + p^2(\mu - \tilde{X}_{i,k-1})^2\big\rangle \\ &= p(1-p)(m - \mu_{k-1}) + p^2\sigma_{k-1}^2.\end{aligned} \tag{B21}$$

Substitution of Eqs. (B15) and (B21) into Eq. (B14) gives the second formula in Eq. (B1).

□